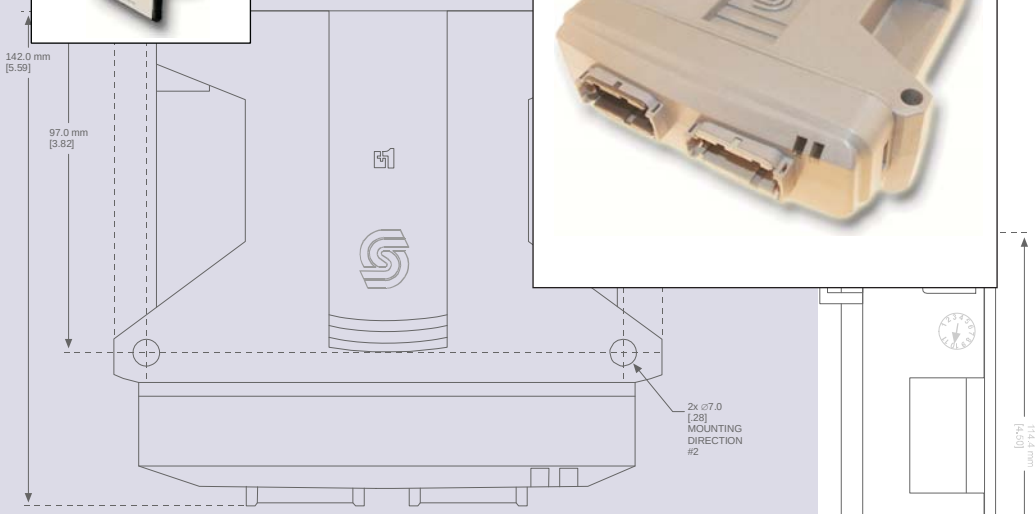
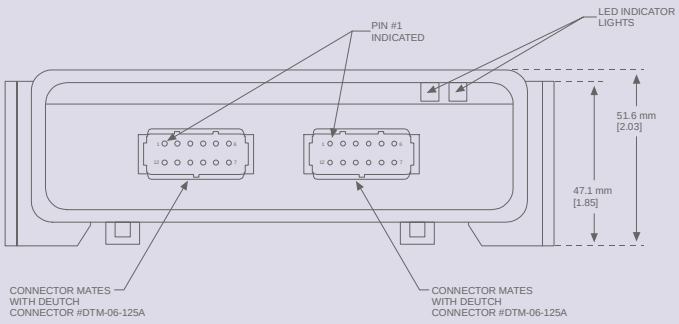
PLUS+1™ GUIDE
Software

PLUS+1 GUIDE
ユーザーマニュアル





更新履歴

Revision	更新日	コメント
Rev FG	April 2009	PLUS+1 GUIDE release 4.2.x.x をサポート

©2009 Sauer-Danfoss. All rights reserved.
ザウアーダンフォスグループは、カタログ・資料およびその他の印刷物あるいは電子資料に生じ得る誤りに対して責任を負うものではありません。また弊社は予告なく製品を変更する権利を有します。この変更は、すでに合意された仕様の変更を必要とするものでない限り、すでに発注された製品にも適用されます。本資料のすべての商標は該当各社が所有するものです。 Sauer-Danfoss、Sauer-Danfossロゴタイプ、S-icon、PLUS+1™、what really matters is inside® Know-How in Motion™はザウアーダンフォスグループの商標です。無断転載を禁じます。

Contents

リスク低減 9
リスクを低減する設計と試験 10
設計 10
試験 10

PLUS+1 GUIDE プログラムについて学ぶ 21
Lesson 1: アプリケーションを作成する 22
Lesson 2: アプリケーションのダウンロード 34

PLUS+1 GUIDE ウィンドウ 39
PLUS+1 GUIDE ウィンドウのエレメント 40
PLUS+1 GUIDE ウインドウについて 42
PLUS+1 GUIDE アプリケーション・テンプレートについて 44
ルート名について 45
PLUS+1 GUIDE ウィンドウメニュー 46
PLUS+1 GUIDE ウィンドウツールバー 53
Print ウィンドウ 57
Options ウィンドウ—General Settings（一般設定） 58
Options ウィンドウ—Auto Pop-ups Settings（自動ポップアップ設定） 60
Options ウィンドウ—Preview Settings（プレビュー設定） 61
Options ウィンドウ—Shortcuts Settings（ショートカットの設定） 62
Options ウィンドウ—File Association Settings（ファイル関連付け設定） 64
Options ウィンドウ—Warnings（警告表示） 65
Options ウィンドウ—Grid Settings 66
Logical Net ウィンドウ 68
Hardware タブ 69
Search ウィンドウ 71
Project Manager タブ 72
Project Manager タブと Hardware タブについて 74
Page Navigator Tab 76
Component タブ 77
Function タブ 78
My Blocks タブ 79
Inspector タブ 80
Compiler Messages タブ 81
Page Interface Editor ウィンドウ 82
Page Interface Editor ウィンドウメニュー 83
Page Interface Editor ウィンドウツールバー 88
ページ、ページのトップビュー、ページインタフェースエディタのウィンドウについて 91
Page コマンドでページを追加/ページインタフェースエディタウィンドウの表示 93
Basic Page の追加/ページインタフェースエディタウィンドウの表示 94
古いページの変更/ページインタフェースエディタウィンドウの表示 95
Module Viewer ウィンドウ 96
Module Viewer ウィンドウメニュー 97
Module Viewer ウィンドウツールバー 101

Contents

データ型 103
データ型 104
オーバーフロー条件について 105
Time Base データ型について 106
分解能 107
Array（配列）データ型について 108

コンポーネント 109
コンポーネントに関するコンテクストヘルプ画面 110
コンポーネント記述について 111
Mathematical (数学的) 113
Arithmetic（算術演算） 114
Absolute Value（絶対値） 126
Scale（スケーリング） 128
Trigonometric（三角関数） 132
Square Root（平方根） 139
Limit（リミット） 140
Max/Min（最大/最小） 141
Compare（比較） 150
Compare（比較） 151
Compare with Hysteresis（ヒステリシス付き比較） 157
Compare in Window（ウィンドウ内比較） 163
Constant（定数） 165
True/False（真/偽） 166
Digit Autotype（桁オートタイプ） 168
Typed（型付き） 170
Logical（論理） 175
AND2–AND8（論理積 2～論理積 8） 176
OR2–OR8（論理和 2～論理和 8） 177
NOT（否定） 178
XOR（排他的論理和） 179
Bitwise（ビット単位） 180
Shift（シフト） 183
Position（位置） 185
Set/Reset Latch（セット/リセットラッチ） 188
Data Flip-Flop（データフリップフロップ） 190
Switch, Counter, Memory（スイッチ、カウンタ、メモリ） 192
Switch Boolean Controlled（ブール制御スイッチ） 193
Switch（スイッチ） 196
Counter（カウンタ） 201
Value Connect（値の接続） 210
Memory（メモリ） 213
Array（配列） 217
Get Array Element（配列要素の取得） 218
Set Array Element（配列要素の設定） 220
Length of Array（配列の長さ） 222
Delete Array Elements（配列要素の削除） 223
Insert Array Elements（配列要素の挿入） 225

Contents

Copy of Array（配列のコピー） 227
Find Array（配列の検索） 229
Array Constant from File（ファイルからの配列定数） 230
Array Constant from Binary File（バイナリファイルからの配列定数） 232
Data Conversion（データ変換） 234
Decode/Encode（復号化/符号化） 235
Split（分割） 241
Combine（結合） 242
Swap（入れ替え） 243
Retype（型変更） 244
Transition, Time（遷移、時間） 245
Transition（遷移） 246
Delay（遅延） 249
Oscillator（発振器） 254
Pulse（パルス） 256
例: Pulse 257
Time Base（基準時間） 258
Measure Period（周期計測） 260
Get Time μs 261
Connection（接続） 263
Checkpoint（チェックポイント） 264
Set Value（値の設定） 269
Set Pulse 270
Hardware（ハードウェア） 271
CAN 282
Non-Volatile Memory Dynamic（不揮発性動的メモリ） 292
Non-Volatile Memory Input（不揮発性メモリ入力） 298
Module（モジュール） 300
Module Single Wire（モジュール単一ワイヤ） 301
Module Bus（モジュールバス） 304
Call Module（モジュール呼び出し） 306
Manage（管理） 309
Repeat 310
Until 311
Generate Compiler Error on Range（範囲に関するコンパイラエラーの生成） 315
Generate Compiler Error on Not Constant（定数でないことに関するコンパイラエラーの生成） 316
Generate Compiler Error on Type（データ型に関するコンパイラエラーの生成） 317
Predefine Type（事前定義データ型） 318
Access（アクセス） 320
Access App Log Enable 321
Disable Raw Applog Data Readout 322
Accessrights App Log Statistics（アプリケーションログ 統計データへのアクセス権） 323
Accessrights App Log Errors（アプリケーションログ エラーデータへのアクセス権） 325

Contents

Accessrights App Log Others（アプリケーションログ その他のデータへのアクセス権）..... 326
Accessrights History（履歴データへのアクセス権）..... 327
Accessrights Read（読取アクセス権）..... 328
Accessrights Write（書込アクセス権）..... 329
Read-only Parameter（読取専用パラメータ）..... 334
Open Parameter Set 335
Close Parameter Set..... 336
Connection 337
Display（ディスプレイ）..... 341
Define Window 342
Line..... 344
Graphic..... 346
Text Label..... 348
Graphic Label 349
String 350
Text Set 358
Define Areas Page（エリア定義ページ）..... 363
Define Screen Page（スクリーン定義ページ）..... 363
Application Log（アプリケーションログ）..... 364
Language Definition Input（言語定義入力）..... 364
Define Application Log Areas Page 367
Define Application Log Page..... 367
Page（ページ）..... 368
Basic Page（基本ページ）..... 368

スクリーンエディタ 369
スクリーンエディタについて..... 370
スクリーンエディタのエレメント..... 371
Define Areas ページ（エリア定義ページ）..... 373
Define Areas ページ/Inspector タブ 375
Define Screen ページ..... 382
Define Screen ページ / Library アイテムの追加 384
Define Screen ページ / Inspector タブ 385
Define Screen ページ / Image Register（イメージの登録）..... 391
Define Screen ページ / Text Register（文字列の登録）..... 394

アプリケーションデータログ機能 396
アプリケーションデータログ機能の基本的な要素..... 397
Define APPAreas Block..... 398
Define APPAreas ブロック/Inspector タブ 400
Define APPLog ブロック..... 403
Define APPLog Block/Add Texts..... 404
Define APPLog ブロック / Text Register Pane（文字列登録ペイン）..... 405
Define APPLog ブロック / Inspector タブ — データ書き込みプロパティ..... 406
Define APPLog ブロック / Inspector タブ — DataValue プロパティ..... 407
アプリケーションデータログの内容を読み取る方法..... 408

Contents

PLUS+1 の手引き 411
選択 412
間違いの取消し 412
マウスクリックによるズーム操作 413
マウスホイールによるズーム操作 413
キーボードによる拡大・縮小 413
単一アイテムの削除 414
複数アイテムの削除 415
ページの全内容を他のページにコピー 416
信号ーバス接続の削除 418
表示の更新 419
ページ全体の表示 419
右クリック＆ドラッグによる表示のパン 420
右クリックによる表示のパン 420
ボタンによるアプリケーションのナビゲーション 421
Page Navigator タブによるアプリケーションのナビゲーション 421
ペインの表示と非表示 422
ハードウェア記述のインストール 423
Inspector タブによるプロパティの変更 425
ポップアップ編集ウィンドウによるプロパティの変更 426
ページの新規作成 427
View Logical Net の使用 438
Read-only Parameters（読み取り専用パラメータ）ファイルの作成 443
読み取り専用パラメータファイルの新規作成 444
異なる値を持つ読み取り専用パラメータファイルの作成 453
パラメータを変更したアプリケーションのための新規読み取り専用パラメータファイルの作成 457
CSV テンプレートファイルのフォーマットについて 463
Readme LHX ファイルの追加 464
パーツ番号やシリアル番号によるダウンロードの制限 466
アプリケーション内の変数へのサービスツールによるアクセスの制限 468
リンクページの作成 471
リンクページについて 472
リンクページのプロパティ 473
リンクシンボルの位置の変更 474
リンクページのオブジェクトページへの変換 477
リンクページとオブジェクトページで作業する 478
リンクページ内での変更 482
ページ間のリンクを切る 483
子ページ間のリンクを切る 484
ページの Name Space の表示 486
ページビューの無効化（ロック） 489
アクセスプロパティ 491
コメントのフォントと色をカスタマイズする 496
プラグインコードの追加 500

スクリーンエディタの手引き 501
スクリーンエディタプロジェクトの開始 502

Contents

Define Areas ページ/ハードウェアポートの割当 505
Define Areas ページ/追加スクリーン・エリアの割当 506
Define Screen ページ/スクリーン・エリアの名前の変更 507
Define Screen ページ/スクリーン・エリアの削除 508
Define Screen ページ/スクリーン・エリアのプロパティの変更 509
Define Areas ページ/信号により有効化されたスクリーン・エリアの初期化 511
Define Screen ページ/スクリーン・エリアの割当 512
Define Screen ページ/スクリーン・エリアの削除 513
Define Screen ページ/バーグラフの作成 514
Define Screen ページ/テキストグループの作成 515
Define Screen ページ/デフォルト言語設定の使用 517
デフォルト言語の変更方法 517
Define Screen ページ/言語の切換 518
Define Screen Page/Import Translated Text into an Application 519
翻訳が必要な文字列を含む CSV ファイルのエクスポート 519
文字列の翻訳 521
翻訳文字列を含む CSV ファイルのインポート 523
言語の順序について 524
Define Screen ページ/データ値の表示 525
Define Screen ページ/データ値のフォーマット 526
Define Screen ページ/画像リストの表示 528
Define Screen ページ/テキストリストの表示 533
Define Screen ページ/アプリケーションフォントのインストール 538
Define Screen ページ / 文字列のフォントプロパティの切り替え 544
Define Screen ページ / 特殊キャラクタの表示 548
キーストロークコードによる方法 549
キャラクタコードによる方法 552
Define Screen ページ / 非ローマンアルファベットの表示 555
Define Screen ページ/動画表示 560
Define Screen ページ/画面印刷 562

キーボードショートカット 566
Classic キーボードショートカット体系について 567
基本キーボードショートカット 568
応用キーボードショートカット 569

ページレイアウトガイドライン 571
ページガイドライン 572
ポートラベルの略語 574
ポートラベル単位の略語 575

リスク低減

この章では、PLUS+1™ GUIDE™プログラムで開発したアプリケーションについて、人身傷害や機器の損傷のリスクを低減するために、その設計と試験の重要性について説明しています。

この章の内容

リスク低減 9
リスクを低減する設計と試験 10
設計 10
試験 10

リスクを低減する設計と試験

一般に PLUS+1 GUIDE プログラムで作成されるアプリケーションは、トラクター、クレーン、ハーベスタなど重量があり、出力が大きいオフロード機器を制御します。

通常の稼動条件においても、この種の機器を扱うことは常に人身傷害や機器の故障のリスクを伴います。異常な稼動条件では人身傷害や機器の故障のリスクは飛躍的に増大します。

PLUS+1 GUIDE プログラムはこれらのリスクに対抗するような自動的な保護機能は備えていません。またツールはツールソフトウェアのバグ、ツールマニュアルの誤記、ツールのソフトウェアバージョンの非互換性に起因するリスクに対抗する保護機能も備えていません。

設計者はこれらのリスクを低減するようにアプリケーションを設計し、試験しなければなりません。

設計

設計者は PLUS+1 GUIDE アプリケーションを設計する場合、正常および異常な稼動条件におけるリスクを低減させるために必要な障害検知やエラーハンドリングを含むように設計する責任を負います。

以下に、アプリケーションのための障害検知やエラーハンドリングを開発する場合に考慮すべき項目を示します。

- 機器は通常どのように使用されるか
- 考えられる操作ミスとその結果
- 業界の安全規格と法的要求項目
- 入力・出力の障害とそれがもたらす結果。これらの障害は以下のものを含みます。
  - ジョイスティック、センサー、その他の入力で、その値が急に 100%や 0%に変化するもの
  - 機器の方向、速度、力を制御する出力で、その方向が急に変化するもの、あるいは値が急に 100%や 0%に変化するもの。

    それぞれの障害がどの程度の可能性で起こるかを決定してください。障害の可能性が高いほどその障害の結果に対する保護を十分行う必要があります。
- 事象のシーケンスと障害やエラーの結果
- 事象のシーケンスと非常停止の結果

試験

アプリケーションを作成した設計者は、そのアプリケーションを試験する責任を負います。

設計者はアプリケーションをハードウェアにダウンロードし、その動作を正常および異常な稼動条件で試験する必要があります。以下のことを確認してください：

- 個々の入力が期待した出力を行うかどうか
- 障害への対処とエラーチェックが設計どおりに動作するかどうか

IEC 61508 SIL2 Certified Code Generation Tool

この章では、PLUS+1 GUIDE プログラムを SIL2 IEC61508 で認証されたコード生成ツールとして使用することについて、英語の原文のまま記述しています。

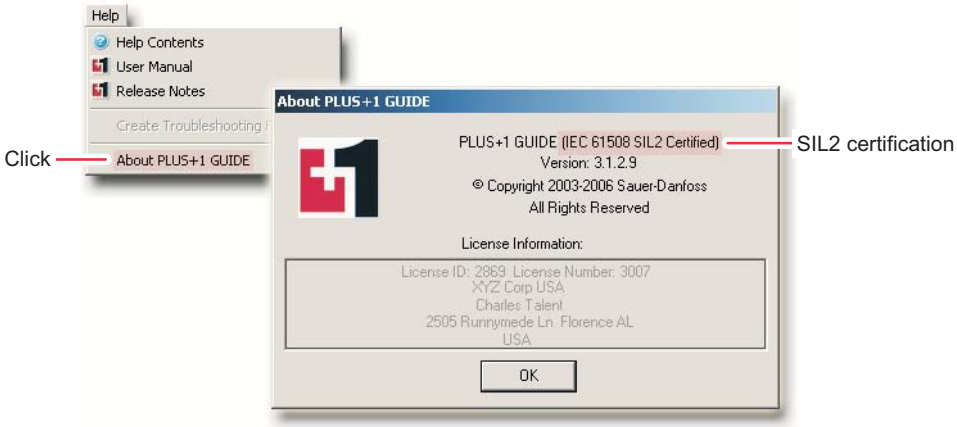
Chapter contents

About SIL2 Certification....12
SIL2 Certification Requirements....13
Additional SIL2 Certification Requirements....20

About SIL2 Certification

The PLUS+1 GUIDE program and the PLUS+1 Service Tool program fulfill the requirements of International Electrotechnical Commission (IEC) standard 61508 for use as support tools in the development of Safety and Integrity Level 2 (SIL2) application software. (Refer to SP Technical Research Institute of Sweden certificate No. 46 24 01.)

IEC standard 61508-3 defines all the requirements that must be fulfilled to produce SIL2 application software. By using the PLUS+1 GUIDE program and PLUS+1 Service Tool program during the application software design, some of the requirements in IEC 61508-3 will be fulfilled. For a more detailed description concerning which of the IEC 61508-3 requirements that are fulfilled or not, see the ***SIL2 Certification Requirements*** on page ***13***.



The **About PLUS+1 GUIDE** window indicates if the version of the PLUS+1 GUIDE program that you are using is SIL2 certified.

Warning

Changing files in a PLUS+1 GUIDE project with any program other than the PLUS+1 GUIDE program can produce an application that operates in an unpredictable and possibly dangerous manner. It also voids SIL2 certification for the application.

Only use the PLUS+1 GUIDE program to make changes to files in a PLUS+1 GUIDE project.

SIL2 Certification Requirements

The following tables:

- Identify, clause by clause, the software quality management system and software safety lifecycle requirements in the 61508-3 standard that must be fulfilled to produce a SIL2 certified application.
- Show the requirements that are either completely fulfilled or partially fulfilled by using the PLUS+1 GUIDE program and PLUS+1 GUIDE Service Tool program.
- Also see ***Additional SIL2 Certification Requirements*** on page ***20***.

6.0 Software quality management system

Clause	Fulfill	Comments
6	No	The PLUS+1 GUIDE software tool does not handle the necessary software quality management system.

7.0 Software safety life cycle requirement

Clause	Fulfill	Comments
7.0	No	The software safety life cycle requirements are partly fulfilled by using the PLUS+1 GUIDE software tool.

7.1 General

Clause	Fulfill	Comments
7.1	No	This requirement structures the development of the software into defined phases and activities. The PLUS+1 GUIDE software tool does not fulfill this requirement.

7.2 Software safety requirements specification

Clause	Fulfill	Comments
7.2	No	These requirements are related to the application created with the PLUS+1 GUIDE software tool. The PLUS+1 GUIDE software tool does not fulfill this requirement.

7.3 Software safety validation planning

Clause	Fulfilled	Comments
7.3	No	These requirements are related to the application created with the PLUS+1 GUIDE software tool. The PLUS+1 GUIDE software tool does not fulfill this requirement.

7.4.2 General requirements

Clause	Fulfill	Comments
7.4.2.1	See clauses 7.4.2.2–7.4.2.12	The user of the PLUS+1 GUIDE software tool and the software tool itself share the responsibility for software design and development. This table describes how responsibilities for the requirements in clause 7.4.2 divide between the user of the PLUS+1 GUIDE software tool and the software tool itself.
7.4.2.2	Yes	The PLUS+1 GUIDE software tool possesses features that facilitate references (a) to (d) as described in clause 7.4.2.2. The software application designer has the responsibility to utilize these possibilities.
7.4.2.3	Partial	The PLUS+1 GUIDE software tool and the PLUS+1 GUIDE Service Tool program provide sufficient means to enable testability and capacity for safe modifications as a design aspect. The software application designer has the responsibility to utilize these possibilities.
7.4.2.4	Yes	The graphical coding environment in the PLUS+1 GUIDE software tool includes features that facilitate software modifications. Such features include modularity, information hiding, and encapsulation.
7.4.2.5	Yes	The design representation is interpreted as a high-level design description. The notation is defined as the available graphical components. The PLUS+1 GUIDE User Manual clearly defines these graphical components.
7.4.2.6	No	The software application designer, who uses the PLUS+1 GUIDE software tool, has the responsibility to minimize the safety-related part of the software.
7.4.2.7	Yes*	If the software is to implement both safety and non-safety functions, then all of the software shall be treated as safety related, unless adequate independence between the functions can be demonstrated in the design. ***If the software application designer wishes to separate safety and non-safety functions when using the PLUS+1 GUIDE software tool, the only possibility is to either use more than one hardware unit in the system or put the non-safety related part in a separate module.***
7.4.2.8	Yes*	If the software is to implement safety functions of different safety integrity levels, then all of the software shall be treated as belonging to the highest safety integrity level, unless adequate independence between the safety functions of the different safety integrity levels can be shown in the design. The justification for independence shall be documented. ***If the software application designer wishes to separate safety functions with different safety integrity levels when using the PLUS+1 GUIDE software tool, the user only has the possibility to put the non-safety related part in a separate module.***
7.4.2.9	No	The software application designer has the responsibility to include software functions to execute proof tests and diagnostic tests (these are functional properties).
7.4.2.10	No	The software application designer has the responsibility to include software functions for self-monitoring of control flow and data flow and to take appropriate actions when failures are detected.
7.4.2.11 (1)	N/A	Handwritten C code is not allowed to link to the PLUS+1 GUIDE software tool.
7.4.2.11 (2)	Yes	The software application designer is responsible for requirements regarding re-use of graphical code. A tested and verified graphical source can be re-used without testing when moved from another hardware platform.
7.4.2.12	No	The software application designer has the responsibility to identify any data (software parameters, for example) influencing the application.

7.4.3 Requirements for software architecture

Clause	Fulfill	Comments
7.4.3.1	See clauses 7.4.3.2–7.4.3.3	The user of the PLUS+1 GUIDE software tool and the software tool itself share the responsibility for software design and development. Clause 7.4.3 is divided between the user and the tool. This table describes the responsibility for handling these requirements.
7.4.3.2	No	The software application designer has the responsibility to develop a description of the software design architecture. However, it is possible to extract relevant parts of the application program architecture description from the PLUS+1 GUIDE software tool.
7.4.3.3	No	The software application designer has the responsibility to decide whether any changes are required to specified safety requirements and to agree to these changes with the hardware developer.

7.4.4 Requirements for support tools and programming languages

Clause	Fulfill	Comments
7.4.4.1	See clauses 7.4.4.2–7.4.4.6	The user of the PLUS+1 GUIDE software tool and the software tool itself share the responsibility for software design and development. This table describes how the responsibility for handling the requirements in clause 7.4.4 is divided between the user of the tool and the tool itself
7.4.4.2	Partial	The PLUS+1 GUIDE User Manual describes the integrated tools within the PLUS+1 GUIDE software tool.
7.4.4.3	Yes	The tool has been certified. Please refer to SP Technical Research Institute of Sweden certificate No. 46 24 01.
7.4.4.4	No	This requirement does not need to be further considered because the PLUS+1 GUIDE software tool satisfies the requirements in 7.4.4.3.
7.4.4.5	No	Coding standards. The assessor and the software application designer are responsible for this requirement.
7.4.4.6	Partial	The PLUS+1 GUIDE software tool coding guidelines specify that coding standards should use good programming practices and proscribe unsafe language features, but it is the responsibility of the software application designer to use the coding guidelines and specify procedures for source code documentation.

7.4.5 Requirements for detailed design and development

Clause	Fulfill	Comments
7.4.5.1	See clauses 7.4.5.2–7.4.5.5	The user of the PLUS+1 GUIDE software tool and the software tool itself share the responsibility for software design and development. This table describes how the responsibilities for requirements in clause 7.4.5 divide between tithe PLUS+1 GUIDE software tool and the software tool itself.
7.4.5.2	No	Information should be available prior to the start of detailed design. This includes the specification of requirements for software safety, the description of the software architecture design, and a plan for validating the software safety. The software application designer is responsible for this requirement.
7.4.5.3	Partial	The software should be produced to achieve modularity, testability, and the capacity for safe modification. The PLUS+1 GUIDE software tool and the PLUS+1 GUIDE Service Tool provide sufficient means to enable testability and capacity for safe modifications as a design aspect. The software application designer has the responsibility to utilize these possibilities.
7.4.5.4	Partial	For each major component/subsystem in the description of the software architecture design, further refinement of the design shall be based on a partitioning into software modules (for example, the specification of the software system design). The design of each software module and the tests to be applied to each software module shall be specified. The PLUS+1 GUIDE software tool supports modular design of the application software. The PLUS+1 GUIDE software tool and the PLUS+1 GUIDE Service Tool provide sufficient means to enable the testing of individual software modules. The software application designer is responsible for specifying the design of each software module and for the tests to be applied to each software module.
7.4.5.5	Partial	Appropriate software system integration tests should be specified to ensure that the software system satisfies the specified requirements for software safety at the required safety integrity level. The PLUS+1 GUIDE software tool and the PLUS+1 GUIDE Service Tool program provide sufficient means to enable integration tests. The software application designer is responsible for specifying appropriate software system integration tests.
7.4.6.1	Partial	The PLUS+1 GUIDE software tool and the PLUS+1 GUIDE Service Tool program provide sufficient means to enable testability and capacity for safe modifications as a design aspect to enable the requirements of (a) to (b) of this clause. The software application designer is responsible for implementing the source code requirements as described in sections (a) to (d) of this clause.
7.4.6.2	No	Each module of software code should be reviewed. The assessor and software application designer are responsible for this requirement.

7.4.7 Requirements for software module testing

Clause	Fulfill	Comments
7.4.7.1	Partial	Each software module shall be tested as specified during software design. The PLUS+1 GUIDE software tool and the PLUS+1 GUIDE Service Tool program provide sufficient means to enable the testing of individual software modules. The software application designer is responsible for testing each software module.
7.4.7.2	Partial	These tests shall show that each software module performs its intended function and does not perform unintended functions. The PLUS+1 GUIDE software tool has flexible means to support modular testing of functions. The software application designer is responsible for the testing requirements.
7.4.7.3	No	The results of the software module testing shall be documented. The software application designer is responsible for this requirement.
7.4.7.4	No	The procedures for corrective action on failure of test shall be specified. The software application designer is responsible for this requirement.

7.4.8 Requirements for software integration testing

Clause	Fulfill	Comments
7.4.8.1	No	Software integration tests shall be specified concurrently during the design and development phase. The software application designer is responsible for this requirement.
7.4.8.2	No	The specified software integration tests shall specify a number of items. The software application designer is responsible for this requirement.
7.4.8.3	Partial	The software shall be tested in accordance with the specified software integration tests. These tests shall show that all software modules and software components/ subsystems interact correctly. The PLUS+1 GUIDE software tool and the PLUS+1 GUIDE Service Tool program provide sufficient means to enable the integration tests. The software application designer is responsible for performing the integration tests.
7.4.8.4	No	The results of software integration testing shall be documented, stating the test results, and whether the objectives and criteria of the test criteria are met. If there is a failure, the reasons for the failure shall be documented. The software application designer is responsible for this requirement.
7.4.8.5	No	During software integration, any modification or change to the software shall be subject to an impact analysis that shall determine all the software modules impacted, and the necessary re-verification and re-design activities. The software application designer is responsible for this requirement.

7.5.2 Programmable electronics integration (hardware and software)

Clause	Fulfill	Comments
7.5.2.1	No	Integration tests shall be specified during the design and development phase to ensure the compatibility of the hardware and software in the safety-related programmable electronics. The software application designer is responsible for this requirement.
7.5.2.2	No	The integration tests for programmable electronics (hardware and software) shall specify the number of items. The software application designer is responsible for this requirement.
7.5.2.3	No	The specified integration tests for programmable electronics (hardware and software) shall distinguish between those activities which can be carried out by the developer on his premises and those that require access to the user's site. The software application designer is responsible for this requirement.
7.5.2.4	Partial	The specified integration tests for programmable electronics (hardware and software) shall distinguish between a number of activities. The PLUS+1 GUIDE software tool only provides sufficient facilities to perform integration tests according to requirement (a) of this clause.
7.5.2.5	No	The software shall be integrated with the safety-related programmable electronic hardware in accordance with the specified integration tests for programmable electronics (hardware and software). The software application designer is responsible for this requirement.
7.5.2.6	No	During the integration testing of the safety-related programmable electronics (hardware and software), any modification or change to the integrated system shall be subject to an impact analysis which shall determine all software modules impacted, and the necessary reverification activities. The software application designer is responsible for this requirement.

7.6 Software operation and modification procedures

Clause	Fulfill	Comments
7.6	No	The software application designer is responsible for this requirement.

7.7. Software safety validation

Clause	Fulfill	Comments
7.7.2.1	No	If the compliance with the requirements for software safety has already been established as part for the E/E/PE safety-related system (see IEC 61508-2, clause7.7), then the validation need not be repeated. The software application designer is responsible for this requirement.
7.7.2.2	No	The validation activities shall be carried out as specified during software safety validation planning. The software application designer is responsible for this requirement.
7.7.2.3	No	The results of software safety validation shall be documented. The software application designer is responsible for this requirement.
7.7.2.4	No	For each safety function, software safety shall be documented according to results (a) through (f) of this clause. The software application designer is responsible for this requirement.

7.7. Software safety validation

Clause	Fulfill	Comments
7.7.2.5	No	When discrepancies occur between expected and actual results, the analysis made and the decision taken on whether to continue the validation or to issue a change request and return to the earlier part of the development life cycle shall be documented as part of the results of the software safety validation. The software application designer is responsible for this requirement.
7.7.2.6	Partial	The PLUS+1 GUIDE software tool and the PLUS+1 GUIDE Service Tool program shall provide sufficient means to support software validation at the software application level (tests that could be performed are limited to available symbols and the information available in the API) The PLUS+1 GUIDE software tool meets requirement (b) but not (a) or (d) of this clause.
7.7.2.7	No	Software tool qualification requirements are as follows: A. All equipment used for validation shall be qualified according to a specification traceable to an international standard (if available), or to a national standard (if available), or to a well recognized procedure. B. Equipment used for software validation shall be qualified appropriately and any tools used, hardware or software, shall be shown to be suitable for purpose. The software application designer is responsible for this requirement.
7.7.2.8	No	Software validation result requirements are as follows: A. The tests shall show that all of the specified requirements for software safety (see clause7.2) are correctly performed and the software system does not perform unintended functions. B. Test cases and their results shall be documented for subsequent analysis and independent assessment as required by the safety integrity level (see IEC 61508-1, clause 8.2.12) C The documented results of software safety validation shall state either that the software has passed the validation or the reasons for its failure. The software application designer is responsible for this requirement.

7.8 Software modification

Clause	Fulfill	Comments
7.8	No	The user writing the graphical application is responsible for this requirement.

7.9 Software verification

Clause	Fulfill	Comments
7.9	No	The user writing the graphical application is responsible for this requirement.

8.0 Functional safety assessment

Clause	Fulfill	Comments
8.0	No	The assessor of the safety critical product is responsible for this requirement.

Additional SIL2 Certification Requirements

When using the PLUS+1 GUIDE software tool, these additional requirements must also be fulfilled to produce a SIL2 certified application.

1. Minimize, as much as possible, the use of read-only and read/write parameters. Especially minimize the use of parameters that affect application behavior.
2. In any application that uses a downloadable read-only parameters file, the output of the **Parameter Open** component must be used to ensure that the parameters file is correct for the application.
3. Analyze and document if output signals can overflow. Determine what actions should occur in cases of overflow. This applies to both capped and non-capped components.
4. A SIL2 application must use the **SIL2 Certified** component to indicate the fulfillment of requirements for a SIL2 certified application. If the requirements are not fulfilled, the application shall not be started.
5. Analyze and document error handling and how the application implements error handling.
6. Analyze and document that the application minimizes the risk in startup situations that involve initializing variables and outputs.
7. Analyze and document that the application minimizes the risk involved in write delays to non-volatile memory.

PLUS+1 GUIDEプログラムについて学ぶ

本章には 2 つの基本レッスンが含まれており、PLUS+1 GUIDE プログラム並びに PLUS+1 サービスツールプログラムについて学ぶことができます。

この 2 つのレッスンは以下を理解するのに役立ちます。

- PLUS+1 GUIDE プロジェクトの作成方法、オープン方法、保存方法
- **PLUS+1 GUIDE** ウインドウの主な特徴
- PLUS+1 GUIDE テンプレートを使用してアプリケーションを作成する方法
- 基本 PLUS+1 GUIDE のキー操作とアプリケーションコマンド
- PLUS+1 サービスツールプログラムを使用して、アプリケーションを MC24-10 Series コントローラにダウンロードする方法

準備方法

- レッスン 1 を完了するためには、PLUS+1 GUIDE プログラムの完全機能版を PC にインストールする必要があります。（レッスン 1 は独立式レッスンです。レッスン 2 に進む必要はありません。）
- レッスン 2 を完了するためには、以下のハードウェアが必要です。
  - MC024-010 PLUS+1 コントローラ（部品番号 10100899）1 台
  - PLUS+1 CAN/USB ゲートウェイ＆ケーブル（部品番号 1091099）1 台
  - 12～24 Vdc、500 mA の電源 1 個

マウスとキーボードの操作

左クリック　　右クリック

左クリックしたままドラッグ　　右クリックしたままドラッグ

Enter キーボードの Enter キーを押す　　F4 キーボードの F4 キーを押す

L キーボードの L キーを押す　　E キーボードの E キーを押す

レッスンで使用する図には、マウスやキーボードの動作を示す記号を使用します。先頭にある図が、その記号です。

マウスまたはキーボード

たいていの PLUS+1 のタスクは、マウスかキーボードを使用して行えます。説明を簡単にするために、ここでは 1 つのタスクをするのに 1 つの方法だけを示します。

詳細情報

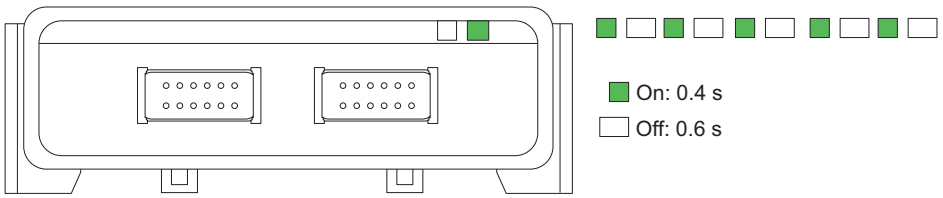
ここに示す基本手順についての詳しい情報は、333 ページの"PLUS+1 の使い方"を参照してください。

本章に示すキャプチャー画面は、お手元の PLUS+1 プログラムの画面表示と少し異なる場合もあります。

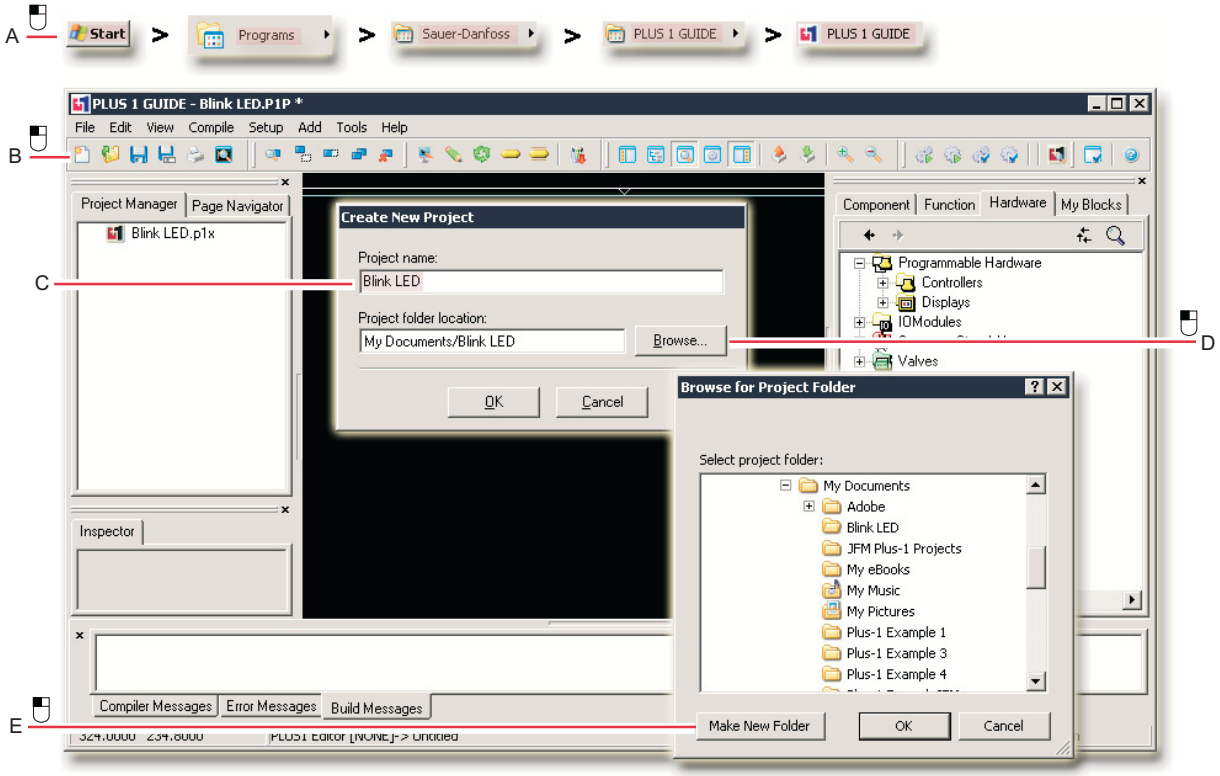
この章の内容

Lesson 1: アプリケーションを作成する 22
Lesson 2: アプリケーションのダウンロード 34

Lesson 1: アプリケーションを作成する



このレッスンでは、アプリケーションを作成し、コンパイルします。ダウンロードすると、MC024-10 コントローラの緑の LED が 0.4 秒間の ON と 0.6 秒間の OFF で点滅します。



1. PLUS+1 GUIDE プログラムを起動し、新規に PLUS+1 プロジェクトフォルダーを作成します。

   A. PLUS+1 GUIDE プログラムを起動します。

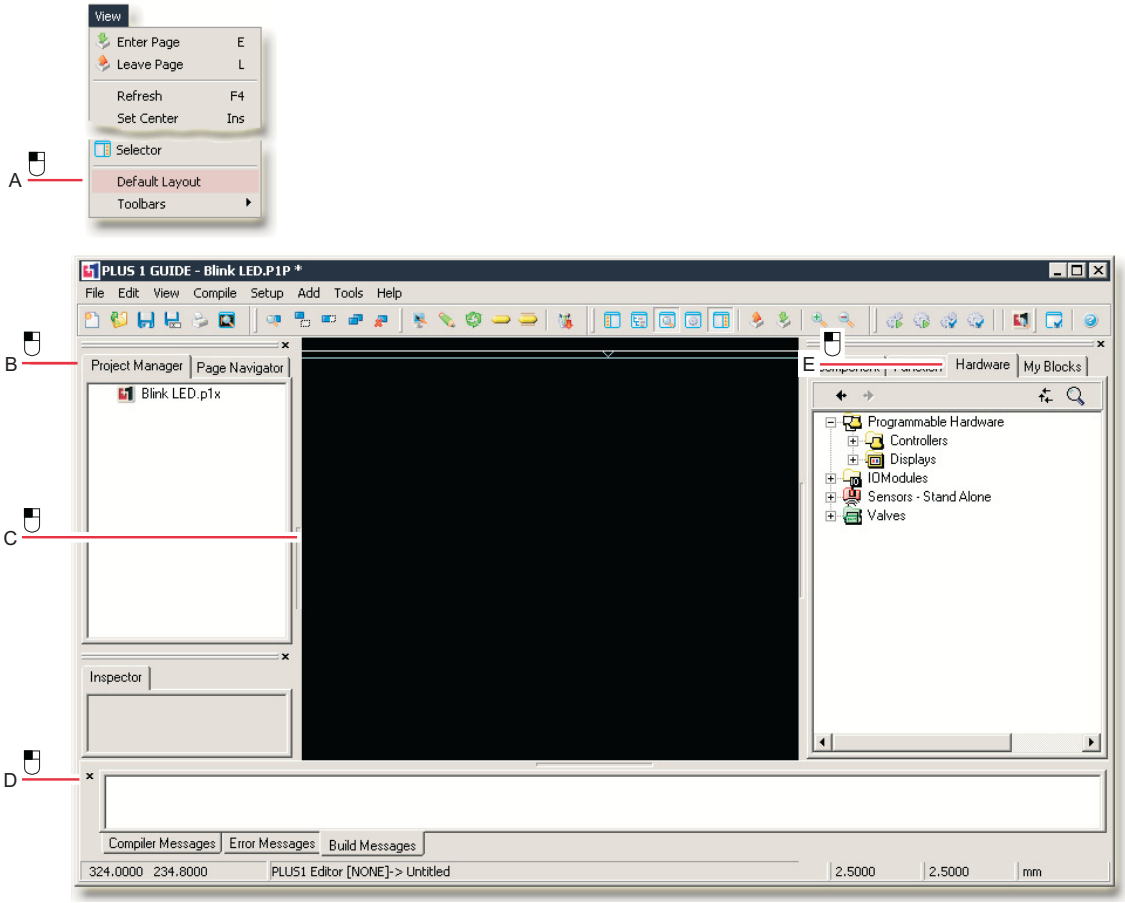
   B. **PLUS+1 GUIDE** ウインドウのツールバーの New Project ボタンをクリックして **Create New Project**（新規プロジェクトの作成）ウインドウを表示させます。

   C. **Create New Project** ウインドウでプロジェクトに **Blink LED** という名前をつけます。

   D. クリックして **Browse for Folder**（フォルダの閲覧）ウインドウを表示させます。

   E. **Browse for Folder** ウインドウを使用して、**Blink LED** という名前のプロジェクトフォルダを作成します。

   各 PLUS+1 プロジェクトはそれぞれ個別のプロジェクトフォルダーを必要とします。



2. **PLUS+1 GUIDE** ウインドウの準備をしてハードウェアファイルと必要なアプリケーション・テンプレートを選択し、MC24-010 コントローラ用アプリケーションを作成します。

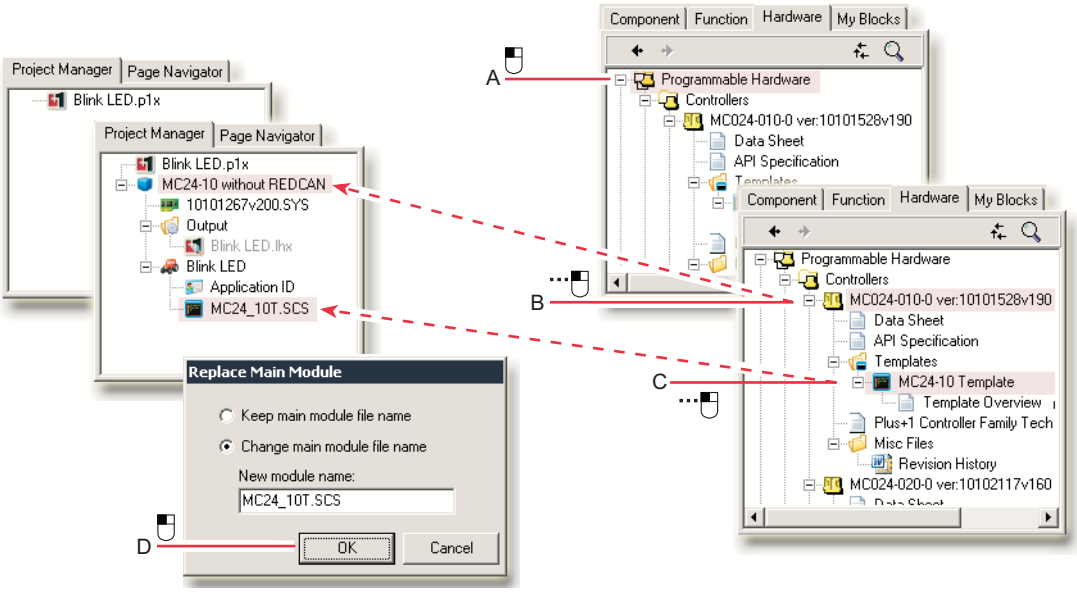
   A. **View**（ビュー）メニューで **Default Layout**（デフォルト・レイアウト）をクリックして、マネージャ、セレクタ、コンパイラのペインを表示します。

   B. **Project Manager**（プロジェクト・マネージャ）をクリックして、このタブを表示します。

   C. クリックして **Inspector** タブを閉じます。

   D. クリックしてコンパイラ・ペインを閉じます。

   E. クリックして **Hardware**（ハードウェア）タブを表示します。



3. MC24-10 ハードウェア説明と **Template** をクリックして、**Hardware** タブから **Project Manager** タブにドラッグします。

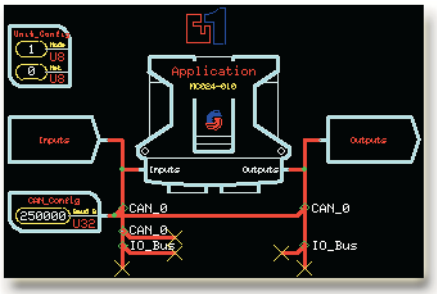
   A. **Hardware** タブで **Programmable Hardware**(プログラム可能ハードウェア) ファイルのツリーを完全に展開し、このツリーの全てのコントローラを表示します。

   B. MC24-10 コントローラのハードウェア記述をクリックして、**Hardware** タブから **Project Manager** タブへドラッグします。

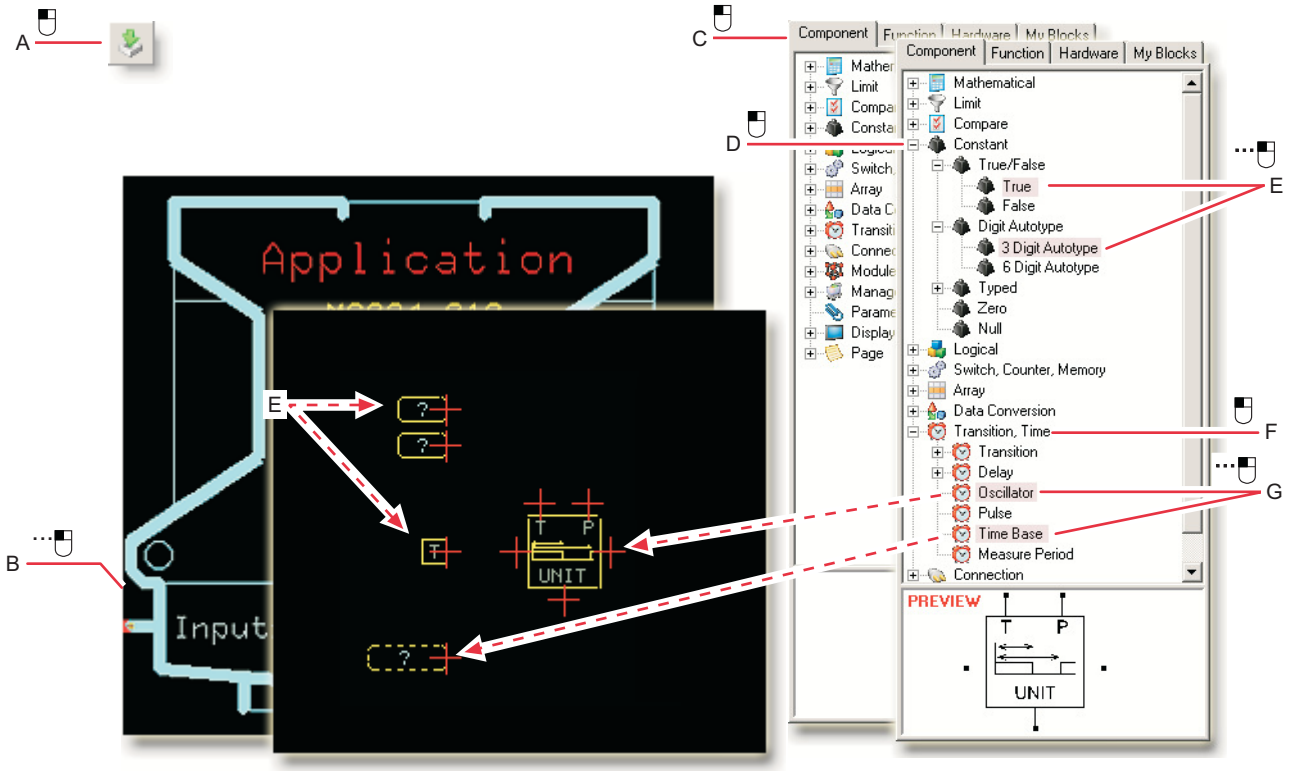
   C. MC24-10 コントローラの **Template** をクリックして、**Hardware** タブから **Project** タブへドラッグします。

   D. **Replace Main Module**(メインモジュールの交換)ウインドウにおいて、**OK** をクリックしてモジュール **MC24_10T.SCS** にリネームします。

MC24-10 コントローラ テンプレート



図の部分は、MC24-10 コントローラ用アプリケーションを作成するためのテンプレートを表示しています。このテンプレートは、MC24-10 コントローラのハードウェアに合う入力装置・出力装置・メモリ位置を予め定義しています。(このテンプレートの別のリリース版の場合、表示は少し異なります。)



4. テンプレートの **Application** ページに入ります。オシレータを構築するのに必要なコンポーネントを選択します。

   A. ツールバーで、Enter Page（ページに入る）ボタンをクリックします。

   B. **Application** ページのポートのいずれかをドラッグするか、ページ境界の内側をクリックして、ページに入ります。

   C. **Component**（コンポーネント）タブをクリックして表示します。

   Component タブは機能別にコンポーネントを分類しています。

   D. クリックして全ての **Constant**（定数）コンポーネントを表示します。

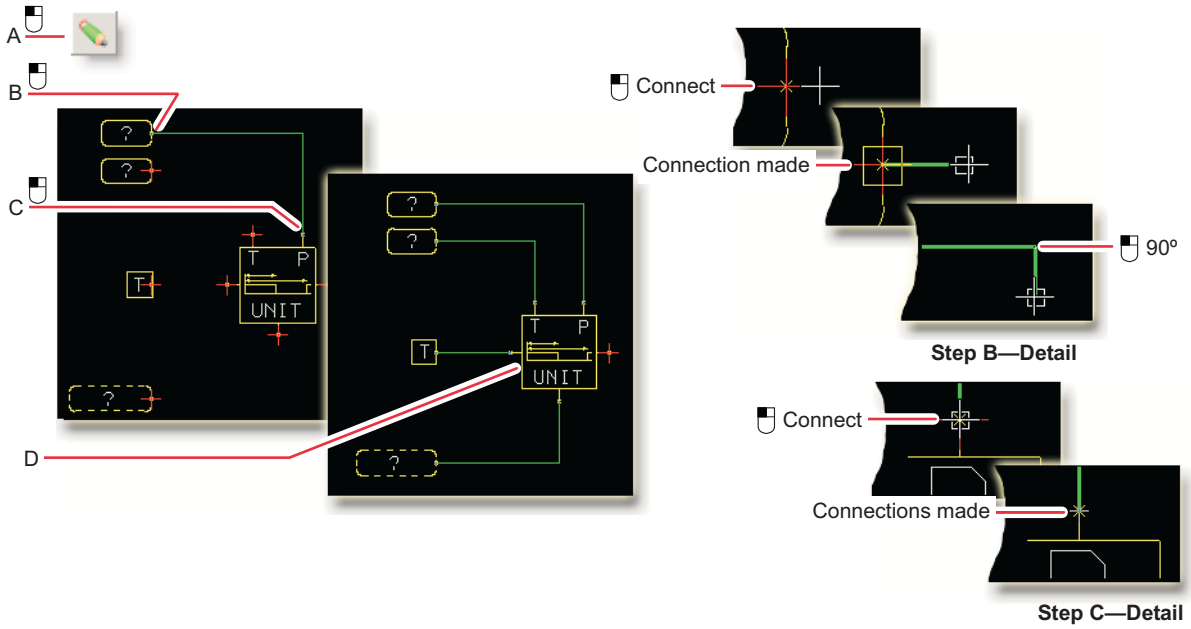
   E. 以下の 2 つの **Constant** コンポーネントを **Application** ページにドラッグします。

   - **3 Digit Auto-type**（3 桁の自動型定数）（2 こ）
   - **True**（真）

   F. クリックして全ての **Transition and Time**（遷移と時間）コンポーネントを表示します。

   G. この 2 つの **Transition and Time** コンポーネントを **Application** ページにドラッグします。

   - **Oscillator**（オシレータ）
   - **Time Base**（時間基準）



5. **Application** ページに置いたコンポーネントを互いに接続します。

   作業に応じて拡大･縮小して見ることができます。

   A. ツールバーで、Route Wire/Bus ボタンをクリックします。

   B. クリックして、一番上の **3 Digit Auto-type** の定数から結線を始めます。

      必要に応じて、Space を押して強制的に接続します。

      黄色い四角のハイライト部分は、たった今、接続された箇所です。

      ワイヤ上のより小さな白い四角は、このワイヤを **Oscillator** の **P**（周期）ピンに接続する場所を示しています。

      ワイヤをクリックして 90°の角度で曲げます。

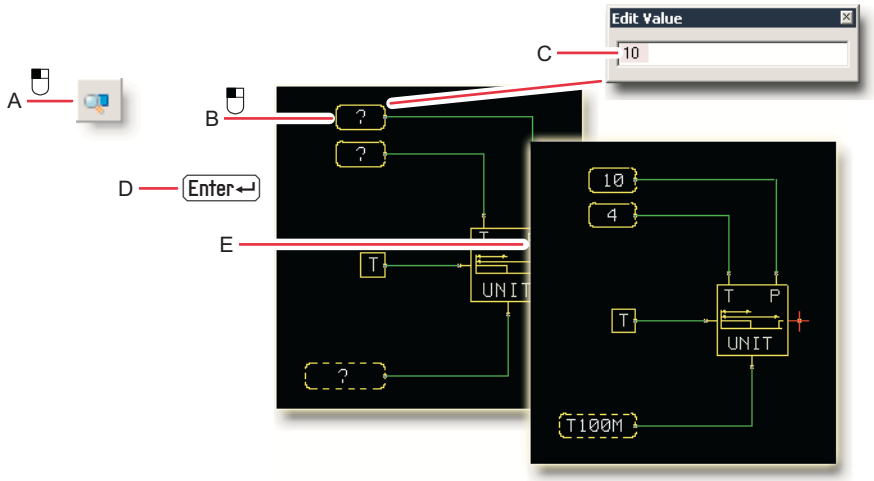
   C. クリックしてワイヤを **Oscillator** の **P** ピンに接続します。

      小さな星印(*)がワイヤの両端に現れ、2 つのコンポーネント間が正しく接続されていることを示します。

   D. 他のコンポーネントをこの図に示した結線のとおりにワイヤで接続します。

      （このレッスンでは真っ直ぐに結線する必要はありません。）

次のステップは、**Edit Value**（値の編集）ポップアップウインドウの機能を使用します。この機能はデフォルトでは有効になっていますが、**Options**（オプション）ウインドウを使用して、有効と無効を切り替えることができます。
Option ウインドウへのパスは、**Setup**（セットアップ）メニュー＞**Options**＞**Options** ウインドウ＞**General**（一般）＞**Pop-up Editor window for components**（コンポーネント用ポップアップ編集）のチェックとなっています。



6. **Edit Value** ウインドウを使用して 3 Digit Auto-type 定数と Time Base 定数に数値を与えます。

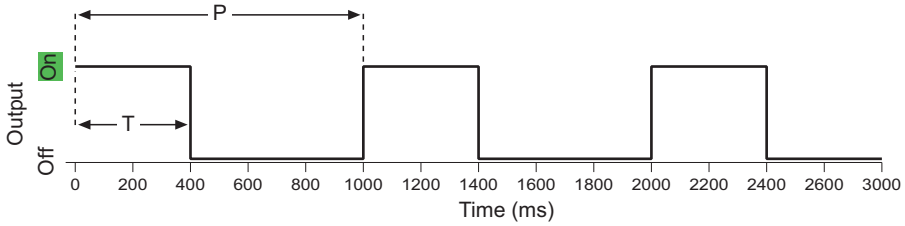
   A. ツールバーの **Query/Change** ボタンをクリックします。

   B. 一番上の定数の中にある **?** をクリックして **Edit Value** ウインドウを表示します。

   C. 表示された **Edit Value** ウインドウで、**10** を入力します。

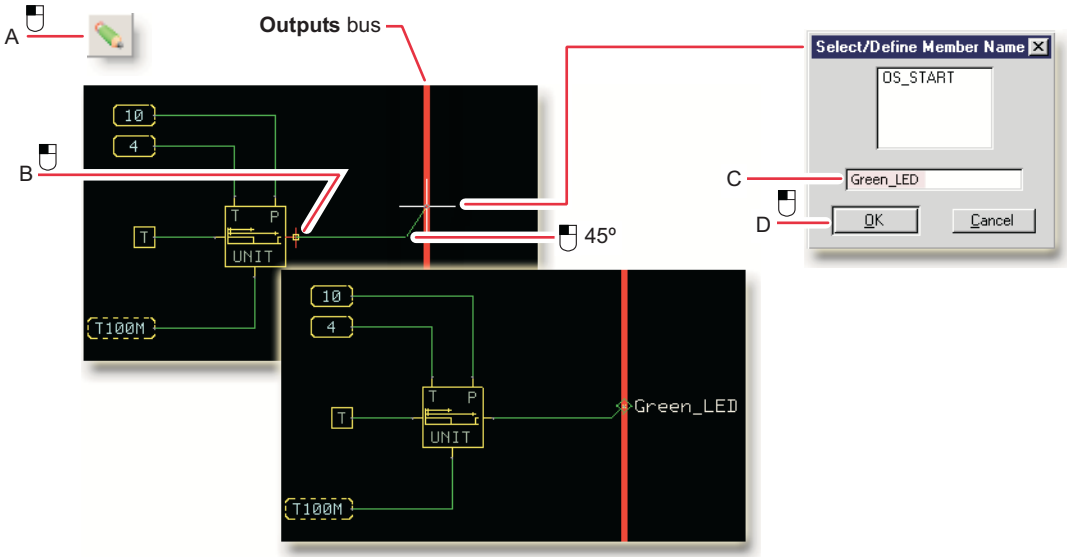
   D. Enter を押して、この定数に **10** を設定します。

   E. 上の図に示された他の定数に値を設定します。



このステップで入力した値が上の図で示されたオシレータの出力になります。

- **Oscillator** の **Unit** 入力に設定した **T100M** によって、タイムベースは 100 ms になります。
- **Oscillator** の **T**（時間）入力に設定した数値 **4** によって、オン時間は 400 ms(4 x 100 ms タイムベース = 400 ms)になります。
- **Oscillator** の **P**(期間）入力に設定した数値 **10** によって、周期は 1000 ms（10 x 100 ms タイムベース = 1000 ms)になります。
- **T**（トゥルー）値によって **Oscillator** 出力が有効になります。



7. **Oscillator** の出力を **Outputs** バスに結線します。

   緑のワイヤは単一信号を伝えます。赤のバスは、主にアプリケーションのページ間で複数の信号を伝えます。

   バスから配線（描画）を開始すると赤のマルチシグナルバスが得られます。単一信号源から配線を開始すると緑の単一信号ワイヤが得られます。

   A. ツールバーで、**Route Wire/Bus** ボタンをクリックします。

   B. **Oscillator** 出力を **Outputs** バスに接続します。

      Oscillator 出力ピンをクリックして結線を開始します。

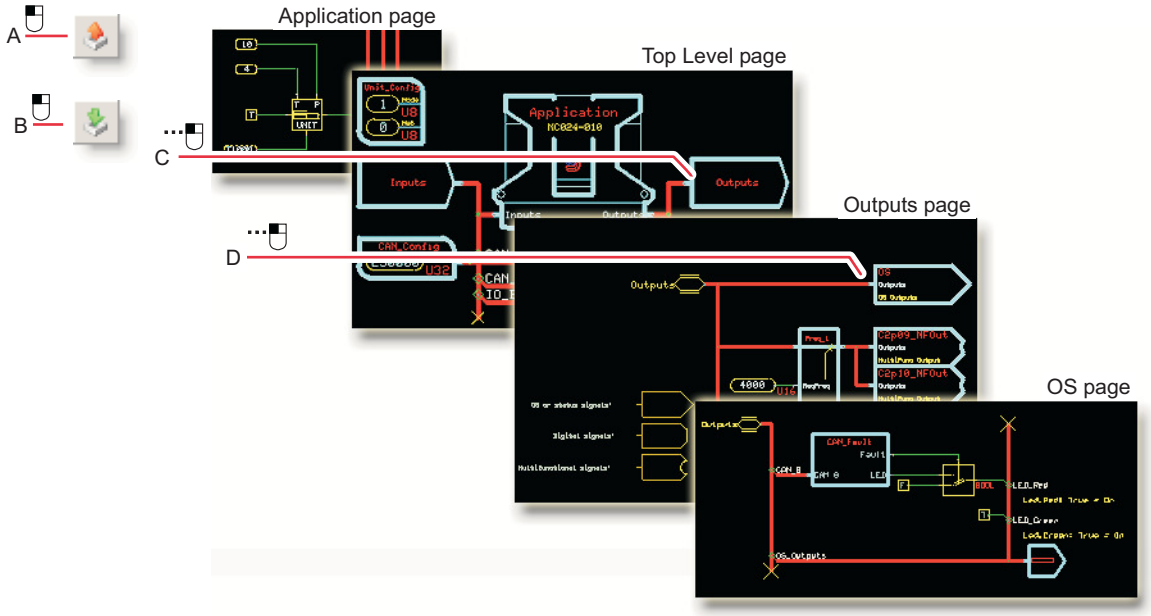
      ワイヤが **Outputs** バスに届く直前にクリックしてワイヤを **Outputs** バスの方向に 45° 曲げます。

45°の角度でワイヤをバスに接続することで、ワイヤの名前を隠さずに済みます。

Outputs バスで結線したときに、**Select/Define Member Name**（メンバー名の選択／定義）ウインドウが現れます。

C. **Select/Define Member Name** ウインドウで、**Oscillator** 出力信号を **Green_LED** と名付けます。

D. **OK** をクリックして、**Select/Define Member Name** ウインドウを閉じます。



8. **OS** ページへのナビゲーション。

OS ページは、**OUTPUTS** ページ内にあるサブページです（代表的なテンプレートページ構造を示す 34 ページの図を参照）。

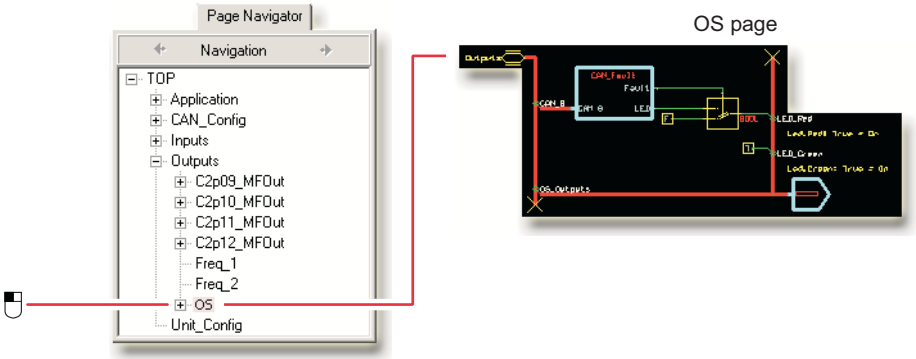
コントローラのオペレーティングシステム（OS）出力に与えた信号が LED を点滅させます。

A. ツールバーで、Leave Page ボタンをクリックして **APPLICATION** ページを離れて、トップレベルページに戻ります。

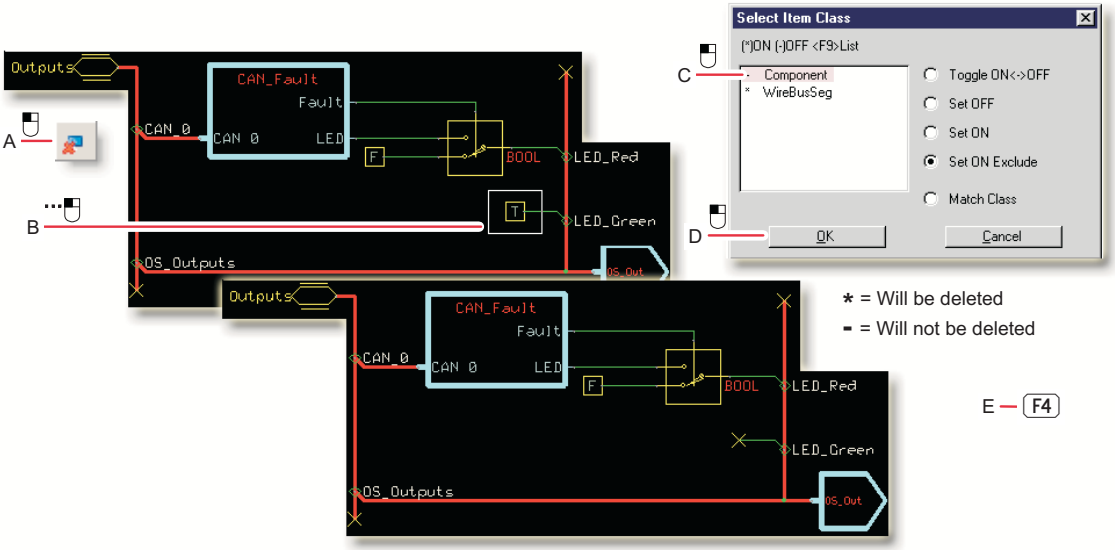
B. ツールバーで、Enter Page ボタンをクリックします。

C. **OUTPUTS** ページのポートをドラッグするか、ページ境界の内側をクリックしてページに入ります。

D. **OS** ページのポートをドラッグするか、ページ境界の内側をクリックして ページに入ります。



Page Navigator タブを使用して、直接 **OS** ページに行くこともできます。



9. **LED_GREEN** に与える定数 **True**（トゥルー）を削除します。

   この **True** によって、コントローラの緑の LED は常に ON となります。

   A. ツールバーで、Delete ボタンをクリックします。

   B. クリックとドラッグによって **True** コンポーネントを選択します。

   マウスから手を離すと **Select Item Class** ウインドウが現れます。

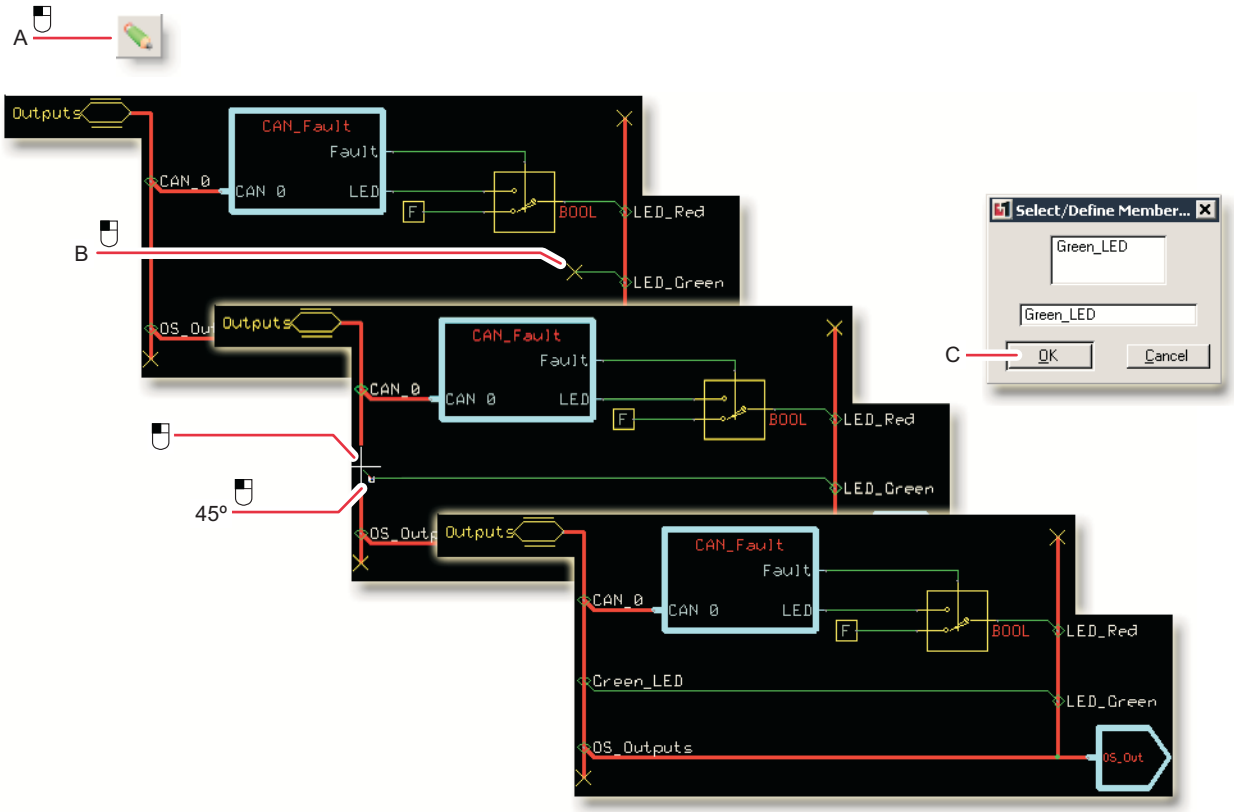
   C. **Select Item Class** ウインドウで、**Toggle ON<->OFF** をクリックして、

   – 星印(*)を **Component** の頭に入れて、**True** コンポーネントを削除するようにします。

   – ダッシュ(–)を **WireBusSeg**（ワイヤバスセグメント）の頭に入れて、ワイヤを削除しないようにします。

   D. **OK** をクリックして、**True** コンポーネントを削除します。

   E. F4 を押して、図面エリアビューをリフレッシュします。



10. **Green_LED** 信号を **LED_GREEN** ポートに配線します。

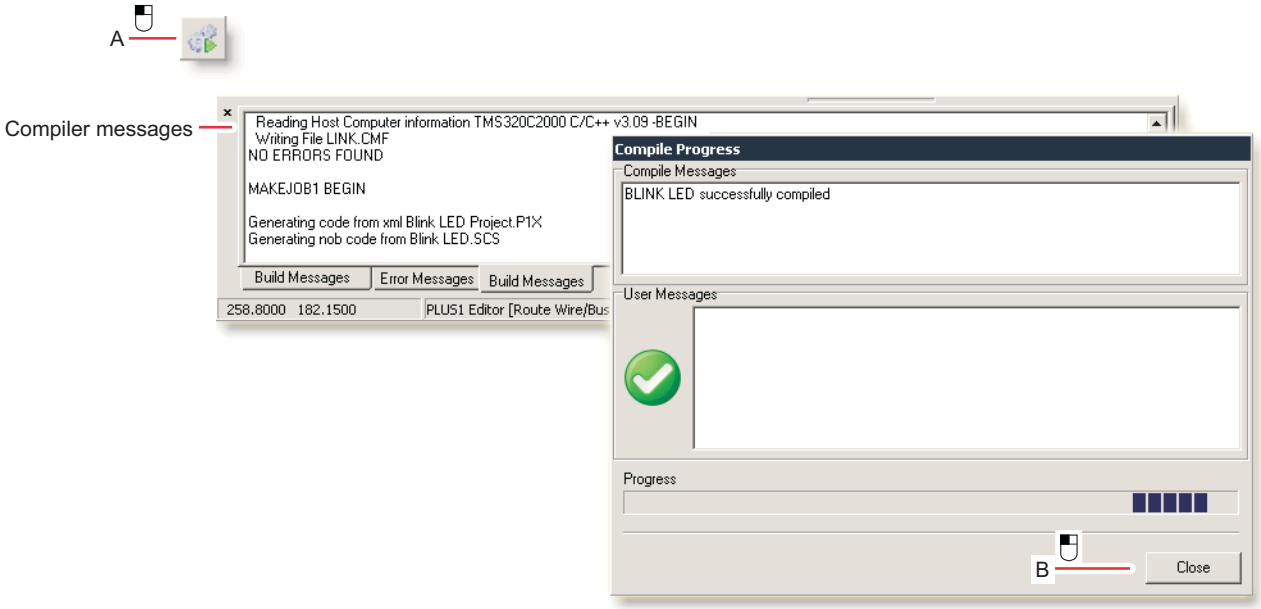
    OUTPUTS バスは **Green_LED** 信号を伝えます。**APPLICATION** ページの **Oscillator** によって、この信号がバスに乗ります。

    A. ツールバーで、Route Wire/Bus ボタンをクリックします。

    B. **LED_GREEN** ワイヤセグメントを **OUTPUTS** バスに接続します。

    - ワイヤの端にある **X** をクリックして、このワイヤを **OUTPUTS** バス方向に引っ張ります。
    - クリックしてワイヤを **OUTPUTS** バスの方向に 45°曲げます。
    - **OUTPUTS** バスをクリックすると、このバスが伝える全ての信号をリストアップした **Select/Define Member Name** ウインドウが現れます。

    C. **Select/Define Member Name** ウインドウで **OK** をクリックして、この信号を **LED_GREEN** ポートに接続します。



11. ダウンロード可能な LHX フォーマットファイルにアプリケーションをコンパイルします。

    A. ツールバーで、Compile All ボタンをクリックして、コンパイル処理を起動します。

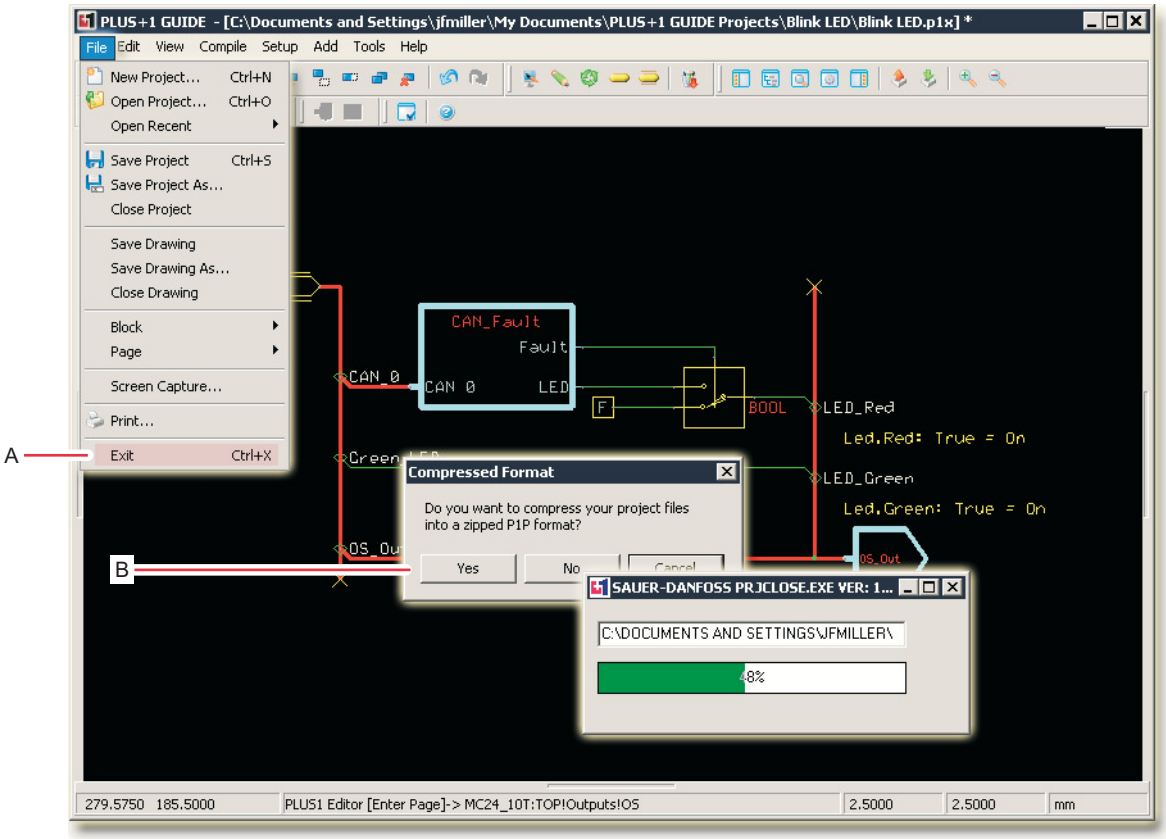
    PLUS+1 GUIDE ウインドウの下にあるコンパイル・ペインが開いて、**Build Messages**（メッセージ作成）タブを表示します。このタブは、コンパイラの進行状況メッセージを表示します。

    Compile Progress（コンパイル進行状況）ウインドウがコンパイル処理の途中に現れ、コンパイル処理の成否を示します。

    B. **OK** をクリックして **Compile Progress** ウインドウを閉じます。

12. 本アプリケーションをダウンロードするのに必要なハードウェアとケーブルがない場合は、33 ページのステップ 13に進んでください。

    本アプリケーションをダウンロードするのに必要なハードウェアとケーブルがある場合は、次ページを飛ばして、***34*** ページの ***Lesson 2:*** *アプリケーションのダウンロード*に進んでください。



13. PLUS+1 GUIDE プログラムを終了します。

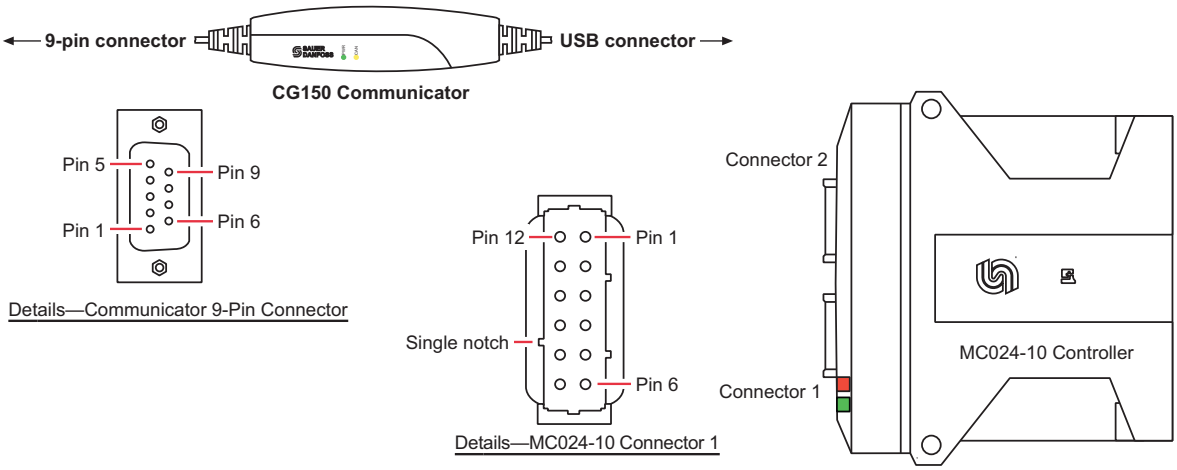
    A. **File** メニューで、**Exit** をクリックします。

    B. **Compressed Format**（圧縮フォーマット）ウインドウで、**Yes** をクリックして、プロジェクトファイルを PLUS+1 Packed (P1P) フォーマットのファイルに圧縮します。

Lesson 2: アプリケーションのダウンロード

PLUS+1 GUIDE プログラムには、PLUS+1 サービスツールプログラムも含まれています。このプログラムは以下の用途に使用します。

- アプリケーションを PLUS+1 プログラマブルハードウェアにダウンロードします。
- パラメータ（動作値）を PLUS+1 プログラマブルハードウェアにダウンロードします。
- PLUS+1 プログラマブルハードウェアの動作を記録します。



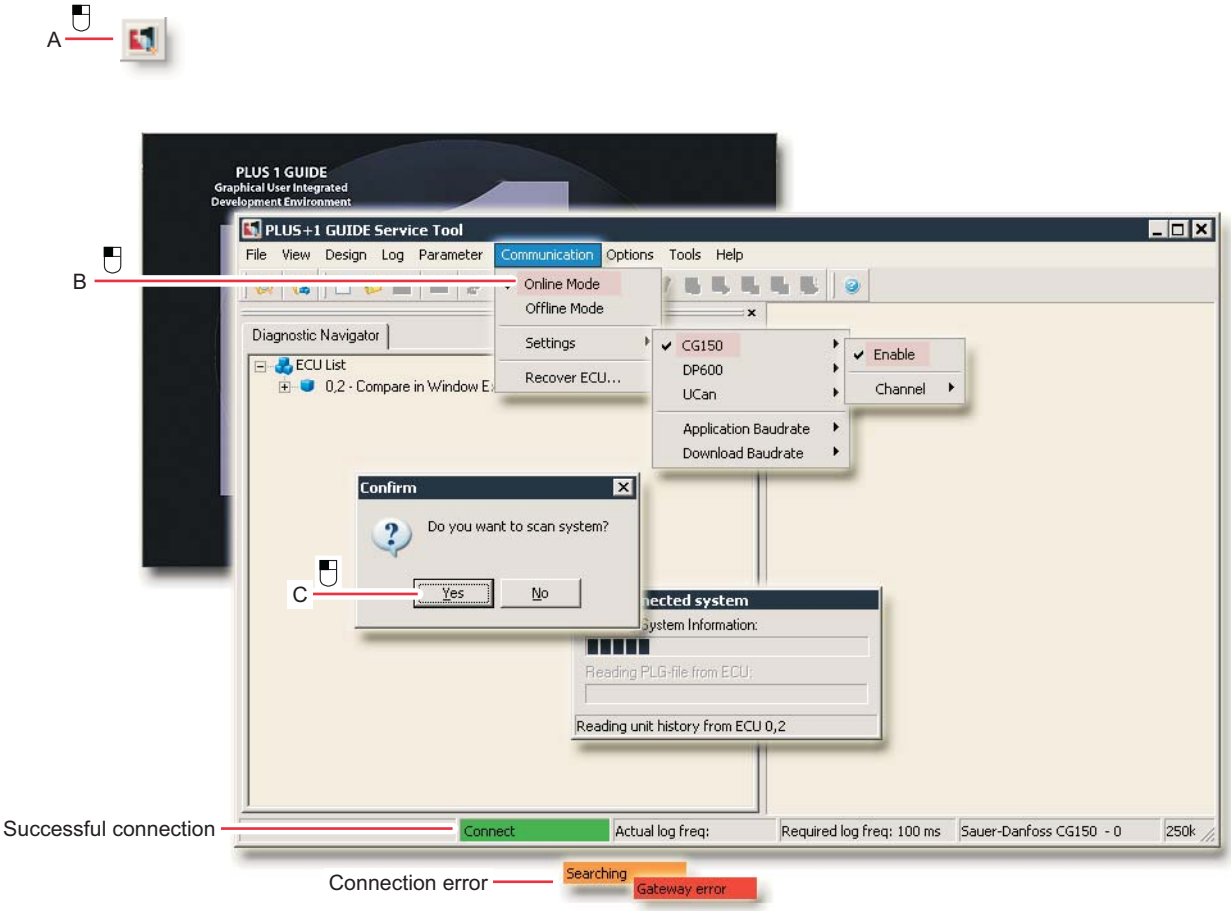
以下の手順では、CG150 CAN/USB ゲートウェイ・インターフェイス・コミュニケータと PLUS+1 MC024-10 コントローラを使用します。

- CG150 の USB コネクタは PC の USB ポートに接続します。
- CG150 コミュニケータと MC-024-10 コントローラの残りの接続を行うときには、以下の表を参考にしてください。

CG150 コミュニケータ／MC024-10 コントローラの接続

機能	CG150 コミュニケータの 9 ピンコネクタ	MC024-10 コントローラコネクタ 1
CAN+	ピン 7	ピン 3
CAN-	ピン 2	ピン 4
CAN シールド/グラウンド	ピン 3	ピン 5
コントローラ電源電圧 +	——	ピン 2
コントローラ電源グラウンド -	——	ピン 1
コントローラ電源グラウンド -	——	ピン 1

1. アプリケーションをコントローラにダウンロードするのに必要なハードウェアをセットアップし、接続します。



2. PLUS+1 サービスツールプログラムを起動します。

   A. **PLUS+1 GUIDE** ウインドウツールバーで、PLUS+1 GUIDE Service Tool ボタンをクリックします。

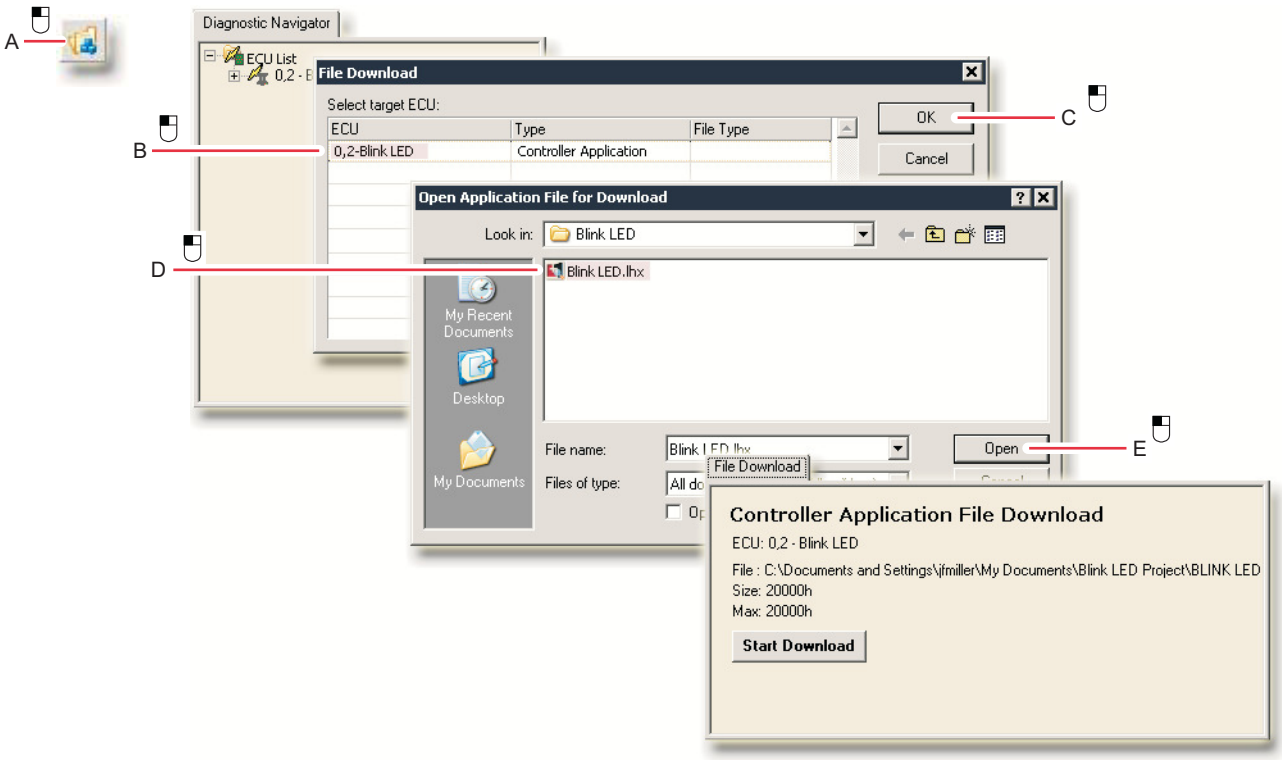
   PC の **Start** メニューからアプリケーションを起動することもできます。

   B. **Communications** メニューで、**Settings** コマンドが **CG150** コミュニケータを選択していることを確認します。

   C. **Confirm**（確認）ウインドウが現れたら、**Yes** をクリックして、ハードウェアをスキャンします。

   スキャンして、ハードウェアメモリ内に保存されたハードウェア情報、アプリケーション情報、履歴情報を読みます。

   PLUS+1 サービスツールウインドウの下にあるステータスバーは、CG150 コミュニケータがコントローラを使って PLUS+1 サービスツールと正しく接続していることを示します。



3. Blink LED.lhx ファイルをコントローラにダウンロードする準備をします。

   A. **Service Tool** ウインドウのツールバーで、File Download ボタンをクリックします。

   このボタンをクリックすると、**File Download** ウィンドウを表示します。

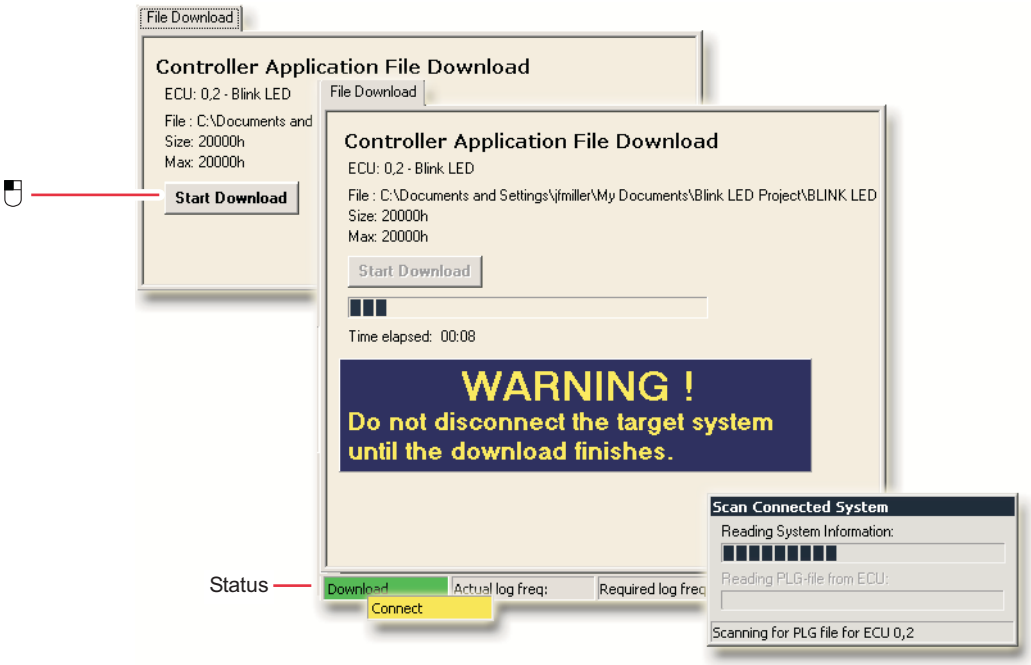
   B. **File Download** ウィンドウで、ターゲット ECU をクリックして選択します。

   （複数のコントローラを使うプロジェクトでは、複数の選択肢が表示されます。）

   C. **File Download** ウインドウで、**OK** をクリックします。

   D. **Open Application File for Download** ウインドウで、**Blink LED.lhx** ファイルをクリックします。

   E. **Open Application File for Download** ウインドウで、**Open** をクリックして、**Blink LED.lhx** ファイルを読み込んでダウンロードします。



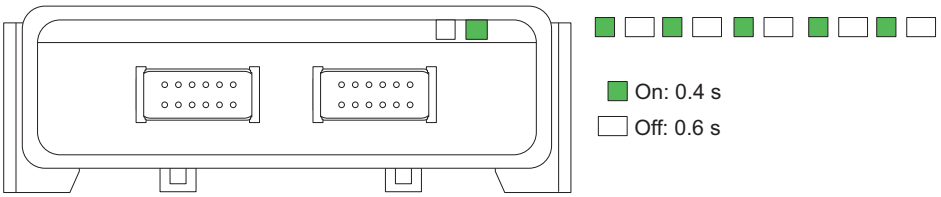
4. **Blink LED.lhx** ファイルをコントローラにダウンロードします。

   File Download タブ内の **Start Download**（ダウンロード開始）ボタンをクリックしてダウンロードを開始します。

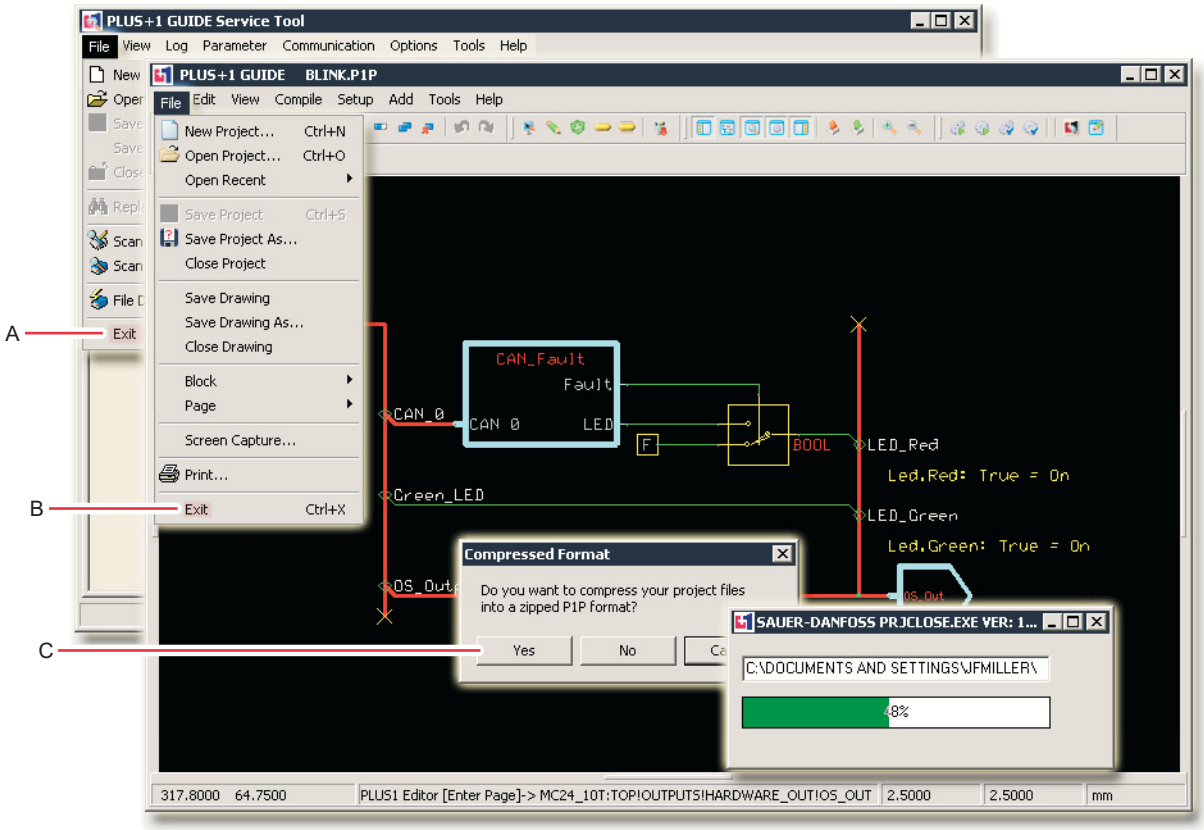
   ウインドウの下にあるステータスバーは以下を表示します。

   - ダウンロード中は **Download** を表示します。
   - ダウンロード前と終了後に **Connect** を表示します。

   PLUS+1 サービスツールプログラムは、ダウンロードが終了すると自動的にコントローラを再スキャンします。



ダウンロードが終了すると、アプリケーションがコントローラの緑の LED の点滅を始めます。



5. PLUS+1 サービスツールプログラムと PLUS+1 GUIDE プログラムを終了します。

   A. PLUS+1 サービスツールウインドウの **File** メニューで、**Exit** をクリックして、PLUS+1 サービスツールプログラムを終了します。

   B. **Exit** をクリックして PLUS+1 GUIDE プログラムを終了します。

   C. **Compressed Format** ウインドウで、**Yes** をクリックして、プロジェクトファイルを一つの PLUS+1 Packed (P1P)フォーマットファイルに圧縮します。

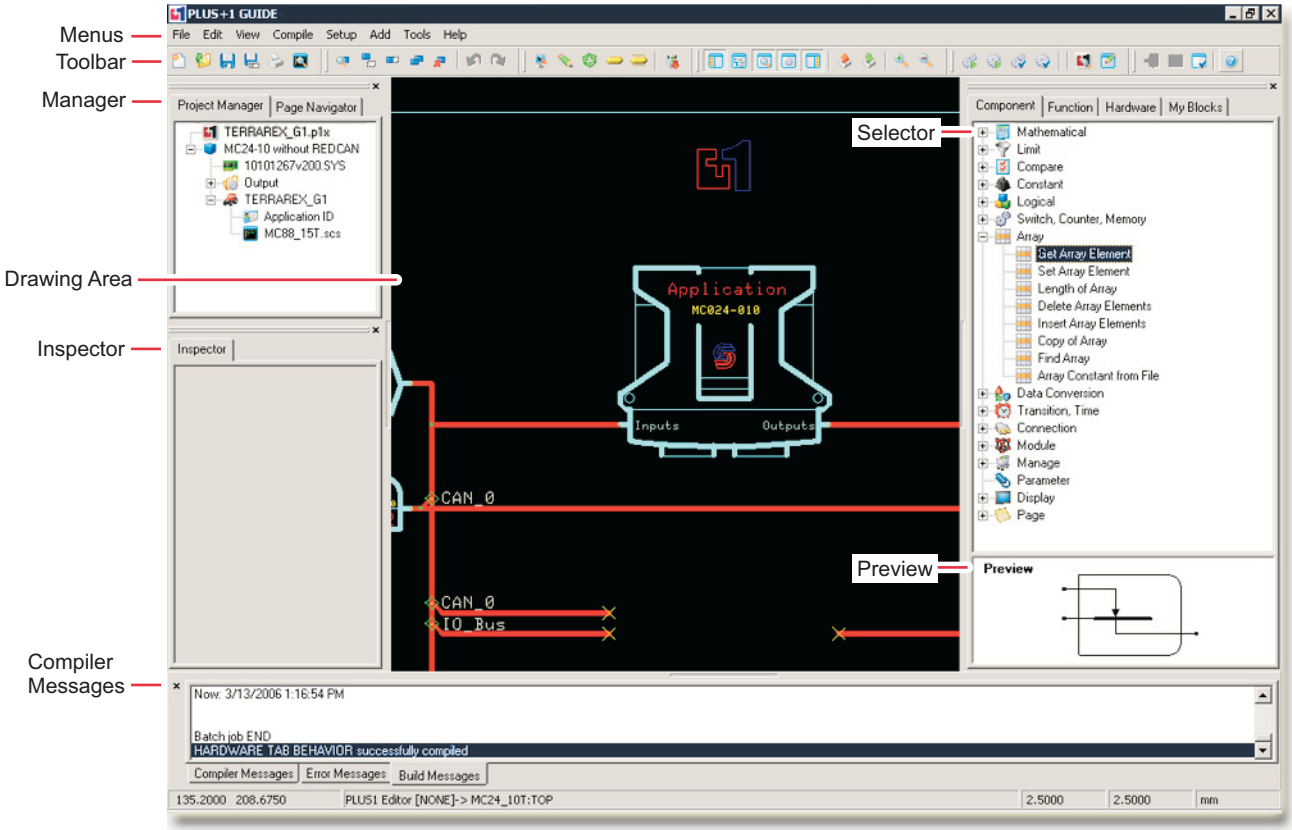
PLUS+1 GUIDE ウィンドウ

本章では、PLUS+1 GUIDE ウインドウの主なエレメントについて説明します。

PLUS+1 GUIDE ウィンドウ....39
PLUS+1 GUIDE ウィンドウのエレメント....40
PLUS+1 GUIDE ウインドウについて....42
PLUS+1 GUIDE アプリケーション・テンプレートについて....44
ルート名について....45
PLUS+1 GUIDE ウィンドウメニュー....46
PLUS+1 GUIDE ウィンドウツールバー....53
Print ウィンドウ....57
Options ウィンドウ—General Settings（一般設定）....58
Options ウィンドウ—Auto Pop-ups Settings（自動ポップアップ設定）....60
Options ウィンドウ—Preview Settings（プレビュー設定）....61
Options ウィンドウ—Shortcuts Settings（ショートカットの設定）....62
Options ウィンドウ—File Association Settings（ファイル関連付け設定）....64
Options ウィンドウ—Warnings（警告表示）....65
Options ウィンドウ—Grid Settings....66
Logical Net ウィンドウ....68
Hardware タブ....69
Search ウィンドウ....71
Project Manager タブ....72
Project Manager タブと Hardware タブについて....74
Page Navigator Tab....76
Component タブ....77
Function タブ....78
My Blocks タブ....79
Inspector タブ....80
Compiler Messages タブ....81
Page Interface Editor ウィンドウ....82
Page Interface Editor ウィンドウメニュー....83
Page Interface Editor ウィンドウツールバー....88
ページ、ページのトップビュー、ページインタフェースエディタのウィンドウについて....91
Page コマンドでページを追加/ページインタフェースエディタウィンドウの表示....93
Basic Page の追加/ページインタフェースエディタウィンドウの表示....94
古いページの変更/ページインタフェースエディタウィンドウの表示....95
Module Viewer ウィンドウ....96
Module Viewer ウィンドウメニュー....97
Module Viewer ウィンドウツールバー....101

PLUS+1 GUIDE ウィンドウ

PLUS+1 GUIDE ウィンドウのエレメント



PLUS+1 GUIDE プログラムを起動すると、このウインドウが現れます。このウインドウを使用して PLUS+1 ハードウェアにダウンロードできるアプリケーションを作成します。

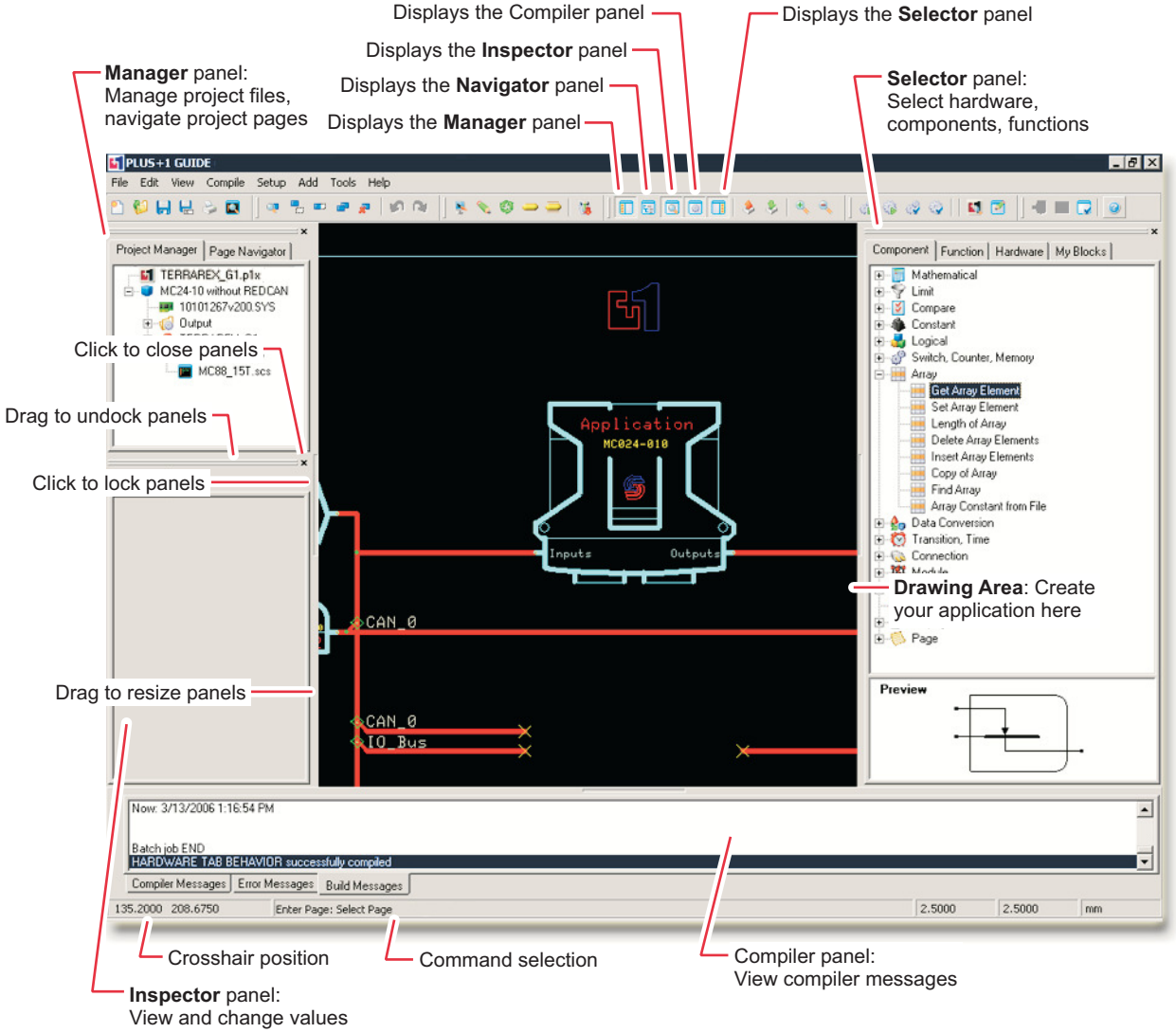
PLUS+1 GUIDE ウインドウのエレメント

項目	説明
メニュー	メニューバーを使用して PLUS+1 GUIDE コマンドとその情報にアクセスします。
ツールバー	ツールバーを使用して、よく使用する PLUS+1 GUIDE コマンドとその情報にアクセスします。
セレクタ	このペインには、**Hardware**（ハードウェア）タブ、**Preview** ペイン付き **Component**（コンポーネント）タブ、**Function**（ファンクション）タブがあります。 **Hardware** タブ： 各 PLUS+1 ハードウェアデバイス用ハードウェア記述のツリービューを示します。このハードウェア記述は PLUS+1 GUIDE プログラムが必要とするリソースにリンクして、特定の PLUS+1 ハードウェアモデルのためのアプリケーションを作成し、コンパイルします。 **Component** タブ： アプリケーション作成時に図面エリアにドラッグできるコンポーネントのツリービューを示します。 **Preview** ペイン： 選択したコンポーネントのプレビューを示します。 **Function** タブ： アプリケーションモジュールの構築時に図面エリアにドラッグできるファンクションのツリービューを示します。
マネージャ	このペインには、**Project Manager** タブと **Page Navigator** タブがあります。 **Project Manager** タブ： **Hardware** タブからドラッグしたハードウェア記述のツリービューを示します。PLUS+1 ハードウェアモデルのためのアプリケーションを作成する前にハードウェア記述をインストールする必要があります。 **Page Navigator** タブ： アプリケーションモジュール内の全ページのツリービューを示します。クリックして選択したページを図面エリアに表示します。

PLUS+1 GUIDE ウインドウのエレメント

項目	説明
図面エリア	アプリケーションモジュールをここで作成します。
インスペクタ	このタブを使用して **Query/Change**（問い合わせ／変更）コマンドで選択したアイテムのプロパティを確認し、変更します。
コンパイラメッセージ	このペインには、**Build Messages**（メッセージ作成）タブ、**Error Messages**(エラーメッセージ）タブ、**Compiler Messages**（コンパイラメッセージ）タブがあります。 **Build Messages** タブ： ダウンロード可能な ***LHX*** フォーマットファイル形式にアプリケーションをコンパイルするときにコンパイラからのメッセージを表示します。 **Error Messages** タブ： コンパイルプロセスでのエラーについての詳細を表示します。 **Compiler Messages** タブ： コンパイラのログファイルに送信したメッセージを表示します。

PLUS+1 GUIDEウインドウについて

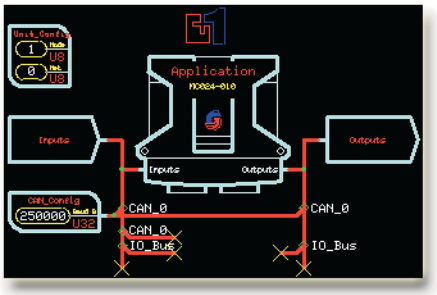


- （このウインドウの右側にある）Selector ペインには、**Hardware**、**Component**、**Function** の各タブがあります。
  - **Hardware** タブを使用して、PLUS+1 の様々なハードウェアモデルに対するアプリケーションを作成・コンパイルするのに必要なファイルを選択します。
  - **Component** タブを使用して、アプリケーション内で使用する **AND** ゲートのような基本コンポーネントを選択します。
  - **Function** タブを使用して、アプリケーション内で使用する **Soft Ramp** のような応用ファンクションを選択します。
- （このウインドウの左側にある）マネージャペインには、**Project Manager**、**Page Navigator** の各タブがあります。
  - **Project Manager** タブは、**Hardware** タブからドラッグしたファイルを表示します。

  - **Page Navigator** は、アプリケーションページのナビゲーションを行います。
- （このウインドウの下部にある）コンパイラペインはコンパイラメッセージを表示します。
- これらのペインは、閉じたり、サイズ変更したり、固定を外したりできます。

PLUS+1 GUIDE アプリケーション・テンプレートについて

アプリケーション・テンプレートのトップレベルページ



アプリケーション **Template** には、論理的にアプリケーション機能を構成したページの階層があります。各 PLUS+1 ハードウェアモデルには、独自の **Template** があります。

細い青のライン(—)は、現在表示しているページの境界を示します。

太い水色のライン(—)は、現在表示しているページ内にネストしているページの境界を示します。

ユーザが独自のページを作成することもできます。347 ページの「ページの作成」を参照してください。

一般的に、コントローラ用アプリケーション **Template** のトップレベルページには、5 つのページがあります。

- **Unit Config**(ユニット構成)ページ: コントロールエリアネットワーク(CAN)**Node**(ノード)と **Net**(ネット)を定義します。
- **CAN Config**(CAN 構成)ページ: CAN ボーレートを設定します。
- **Inputs**(入力)ページ: CAN 通信、コントローラピン入力、OS 入力などの入力機能を定義するサブページがあります。
- **Application**(アプリケーション)ページ: コントローラアプリケーションを作成するページを含みます。
- **Outputs**(出力)ページ: CAN 通信、コントローラピン出力、OS 出力などの出力機能を定義するサブページがあります。

ルート名について

PLUS+1 GUIDE アプリケーション内の全てのルート（緑の単一信号ワイヤと赤のマルチ信号バスの両方）には名前をつける必要があります。

ルートに接続すると、**Select/Define Member Name** ウインドウが現れます。このウインドウに信号名を入れてください。

ルートに名前を付ける時、以下の規則を守ってください。

- 信号名は必ず大文字で始めてください。

正:	誤:	結果:
Green_LED	green_LED	コンパイルエラー

- 信号名を数字で始めないでください。

正:	誤:	結果:
Green_LED_1	1_Green_LED	コンパイルエラー

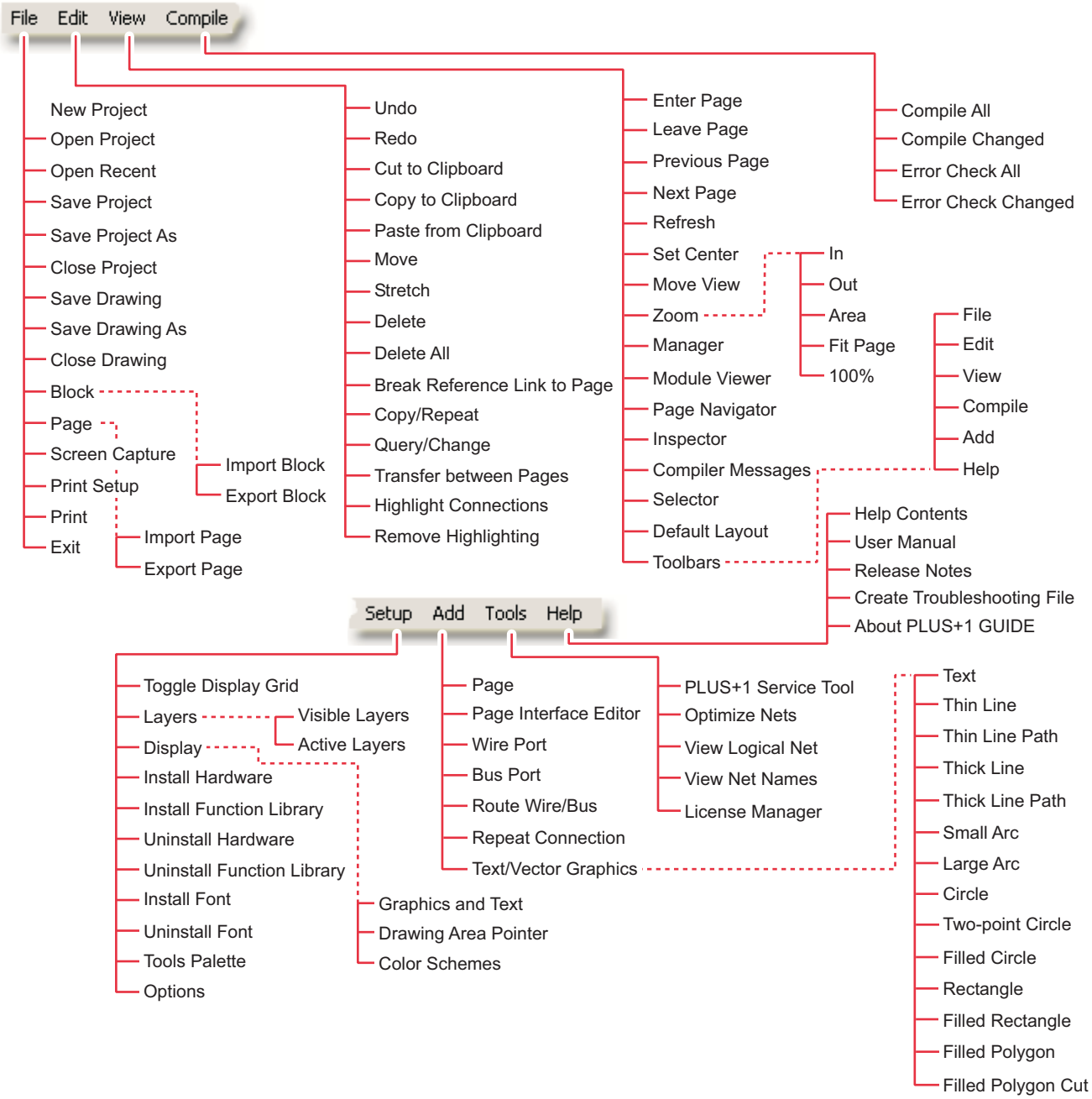
- 信号名のスペースには下線を入れてください。

正:	誤:	結果:
Green_LED	Green LED	コンパイルエラー

- A-Z、a-z、0-9 は、信号名に使用できる文字です。
- 信号名は、大文字と小文字を区別します：Green_LED と GREEN_LED は 2 つの異なる信号を表わします。
- 信号名の長さは 16 文字以下です。

PLUS+1 GUIDE ウィンドウ

PLUS+1 GUIDEウィンドウメニュー



メニューバーを使用して、PLUS+1 GUIDE 機能にアクセスします。

ツールバーボタンは、よく使用するメニューコマンドを抜粋したものです。ツールバーボタンの詳細については。***53***ページ***PLUS+1 GUIDE*** *ウィンドウツールバー*を参照してください。

PLUS+1 GUIDE ウィンドウメニュー

項目	説明
File (ファイル)	このメニューのコマンドはプロジェクトファイルを管理します。
New Project（新しいプロジェクト）	**Create New Project** ウインドウを表示します。 このウインドウを使用して、新しいプロジェクトに名前を付け、プロジェクトファイル用のフォルダを作成します。
Open Project（プロジェクトを開く）	**Open Project** ウインドウを表示します。 このウインドウを使用して、***P1P*** および ***P1X*** プロジェクトファイルを表示して開きます。
Open Recent（最近のファイルを開く）	最近保存されたプロジェクトのリストを表示します。このリストからプロジェクトを選択して、開きます。
Save Project（プロジェクトの保存）	1 つの ***PIP*** フォーマットのファイルに圧縮せずに現在のプロジェクトファイルを保存します。
Save Project As（～としてプロジェクトを保存）	**Save Project As** ウインドウを表示します。 このウインドウを使用して、新しいフォルダ内に新しい名前で現在のプロジェクトを保存します。
Close Project（プロジェクトを閉じる）	PLUS+1 GUIDE プログラムを終了せずに現在のプロジェクトを閉じます。 **Project Modified**（プロジェクト変更）ウインドウと **Save in P1P**（***P1P*** で保存）ウインドウを表示します。 **Project Modified** ウインドウを使用して、プロジェクトで行った変更を保存または廃棄します。 **Save in P1P** ウインドウを使用してプロジェクトファイルを 1 つの ***P1P*** フォーマットのファイルに圧縮します。
Save Drawing（図面の保存）	モジュール名を使用して、アプリケーション図面を SCS ファイルとして保存します。
Save Drawing As（～として図面を保存）	**Select/Define New Job File Name** ウインドウを表示します。 このウインドウを使用して、新しい名前で ***SCS*** ファイルとしてアプリケーション図面を保存します。
Close Drawing（図面を閉じる）	プロジェクトを閉じずに、現在のアプリケーション図面を閉じます。
Block（ブロック）	**Import Block** （ブロックのインポート）: **PLUS+1 Editor Load Block** ウィンドウを表示します。 このウインドウを使用して、***SCS*** ファイルを選択し、PLUS+1 GUIDE ウィンドウの図面エリアにインポートします。 同一ブロックを 2 つ以上インポートするときは、リンク機能によりブロックの内容が自動的にリンクされます。詳細については *472* ページの*リンクページについて*を参照してください。 **Export Block**（ブロックのエクスポート）: **PLUS+1 GUIDE** ウィンドウの図面エリア内でエクスポートするアイテムを選択した後、**Symbol Block Export Binary**（シンボルブロックエクスポートバイナリ）ウインドウを表示します。 このウインドウを使用して、選択したアイテムを ***SCS*** フォーマットのファイルにエクスポートします。 **Export Block** コマンドでは作業中のモジュール内のアイテムしかエクスポートすることが出来ません。外部モジュール、**Module** コンポーネントを通じて参照しているもの（**Call Module** コンポーネントなど）はエクスポートされません。 **Import Block** と **Export Block** コマンドは **Define Areas** と **Define Screen** ブロックに対しては機能しません。
Page（ページ）	**Import Page**（ページのインポート）:**PLUS+1 GUIDE** ウィンドウの図面エリア内でページを選択した後、**Select Job File Name** ウインドウを表示します。 このウインドウを使用して SCS ファイルを選択し、そのコンテンツを選択したページにインポートします。 **Export Page**（ページのエクスポート）: **PLUS+1 GUIDE** ウィンドウの図面エリア内でページを選択した後、**Select/Define New Job File Name**（新しいジョブファイル名の選択／定義）ウインドウを表示します。 このウインドウを使用して、選択したページのコンテンツを SCS ファイルにエクスポートします。 ページ名は 16 文字の制限があるため、SCS ファイル名も 16 文字で制限されます。これより長い名前を持つページはコンパイラエラーを引き起こします。 **Export Page** コマンドでは作業中のモジュール内のページしかエクスポートできません。**Module** コンポーネントを通じて参照している外部モジュール（**Call Module** コンポーネントなど）はエクスポートされません。 **Import Page** と **Export Page** コマンドは **Define Aread** と **Define Screen** ブロックに対しては機能しません。

PLUS+1 GUIDE ウィンドウメニュー

項目	説明
Screen Capture（画面捕捉）	**PLUS+1 GUIDE** ウィンドウの図面エリア内でアイテムを選択した後、**Select Screen Dump Format**（画面ダンプフォーマットの選択）ウインドウを表示します。 このウインドウを使用して、選択したアイテムの印刷、***BMP*** や ***TIFF*** フォーマットのファイルへの保存、あるいは BMP や Metafile フォーマットとしてクリップボードにコピーを行います。
Print Setup（印刷設定）	**Print Setup** ウィンドウを表示します。 このウィンドウを使用してプリンタの選択やそのプロパティの設定を行います。
Print（印刷）	**Print** ウインドウを表示します。 このウインドウを使用してアプリケーションからページを印刷します。
Exit（終了）	PLUS+1 GUIDE アプリケーションを閉じます。 アプリケーションを閉じると、**Project Modified** ウインドウが表示され、続いて **Save in P1P** ウインドウが表示されます。 **Project Modified** ウインドウを使用して、プロジェクトで行った変更を保存するか、または破棄するかを選びます。 **Save in P1P** ウインドウを使用して、プロジェクトファイルを1つの P1P フォーマットのファイルに圧縮します。
Edit (編集)	このメニューのコマンドは、**PLUS+1 GUIDE** ウィンドウの図面エリア内のアイテムを変更・修正します。
Undo（直前の操作の取り消し）	プログラム操作を取り消します。 コンピュータのメモリに応じて、10 個の操作まで取り消しが可能です。
Redo（直前の Undo の取り消し）	**Undo** コマンドを取り消します。 コンピュータのメモリに応じて、10 個までの **Undo** コマンドを取り消せます。 **Option** ウインドウで Undo/Redo 機能の有効化と無効化の切り替えが可能です。このウインドウを表示させるには、**Setup** メニューで **Options** をクリックします。
Cut to Clipboard	**PLUS+1 GUIDE** ウィンドウの図面エリア内で選択したアイテムを削除し、それらをクリップボードにコピーします。
Copy to Clipboard	**PLUS+1 GUIDE** ウィンドウの図面エリア内で選択したアイテムをクリップボードにコピーします。
Paste from Clipboard	クリップボードの内容を **PLUS+1 GUIDE** ウィンドウの図面エリア内に貼り付けます。 リンク機能により、クリップボードからコピーされた複製ページの内容は自動的にリンクされます。詳細な情報については *472* ページ、*リンクページについて*を参照してください。
Move（移動）	**PLUS+1 GUIDE** ウィンドウの図面エリア内で選択したアイテムを移動します。
Stretch（ストレッチ）	選択したルート上のセグメントをストレッチしたり、選択したアイテムを移動したりします。 ストレッチは選択したルートセグメントに頂点を追加します。追加された頂点は、ルート上の他の頂点の位置を変えずに移動させることができます。
Delete（削除）	図面エリアで選択したアイテムを削除します。選択したアイテムは白くなります。 独立したアイテムや同一アイテムを選択すると、**Attributes**（属性）ウインドウが表示されます。**Attributes** ウィンドウ内で **OK** をクリックして選択アイテムを削除します。 **Select Item Class** ウインドウは、複数の異なるアイテムを選択するときに表示されます。このウインドウを使用して、削除したいアイテムを選択します。星印(*)で削除するアイテムを指定し、ダッシュ(-)で削除しないアイテムを指定します。 **Select Item Class** ウィンドウ内で **OK** をクリックして選択アイテムを削除します。
Delete All（全削除）	**Delete all items?**メッセージを含む **Question** ウィンドウを表示します。 **Yes** をクリックして **PLUS+1 GUIDE** ウィンドウの図面エリア内の全てのアイテムを削除します。
Break Reference Link to Page （ページへの参照リンクの切断）	**PLUS+1 GUIDE** ウィンドウの図面エリア内でリンクページを選択したとき、**Break Link** ウィンドウを表示します。 ウィンドウ内の Yes をクリックして、選択したページの他のページへのリンクを切断します。 詳細な情報については *472* ページ、*リンクページについて*を参照してください。

PLUS+1 GUIDE ウィンドウメニュー

項目	説明
Copy/Repeat（コピー／リピート）	PLUS+1 GUIDE ウィンドウの図面エリア内で選択したアイテムをコピーします。選択されたアイテムは白くなります。クリックするとコピーしたアイテムの位置を確定します。Esc でコピーしたアイテムをキャンセルします。 このコマンドは図面エリア内でのみ有効です。 リンク機能により、コピーされたページは自動的にリンクされます。詳細な情報については *472* ページ、*リンクページについて*を参照してください。
Query/Change（問い合わせ／変更）	**PLUS+1 GUIDE** ウィンドウの図面エリア内で、アイテムのプロパティ（データ型、テキスト、機能など）を変更するときに使用します。 プロパティを変更したいアイテムをクリックしてください。選択したアイテムに適したダイアログボックスが表示されます。このウィンドウ内でプロパティを変更します。
Transfer between Pages（ページ間の移行）	元のページの全ての内容をコピー先のページにコピーするのに使用します。コピー先のページはアプリケーション内のどのページでも可能です。元のページをコピー先のページにコピーした後、コピー先のページで作業を続けるか、元のページに戻るかを選択出来ます。詳細な情報については、*416* ページの*ページの全内容を他のページにコピーする*を参照してください。
Highlight Connections（接続のハイライト表示）	ページ内のワイヤ／バス接続をチェックするときに使用します。 セグメントを選択すると、選択したセグメントに接続している全セグメントがハイライトします（白く変ります）。
Remove Highlighting（ハイライト表示の取り消し）	接続したセグメントのハイライト表示を消します。
View（ビュー）	このメニュー内のコマンドは、図面エリアのビューを変更します。
Enter Page（ページに入る）	選択したページに入ります。 ページに入るには、ページ境界の内側をクリックするか、ページのポートをドラッグします。
Leave Page（ページから出る）	現在のページから出ます。
Previous Page (前のページ)	ページナビゲーションの履歴から 1 ページ前に戻ります。
Next Page （次のページ）	ページナビゲーションの履歴から 1 ページ次に進みます。
Refresh（再描画）	**PLUS+1 GUIDE** ウィンドウの図面エリアを再描画します。
Set Center（中央に設定）	ポインタでクリックした場所を PLUS+1 GUIDE ウィンドウの図面エリアの中央に移動します。
Move View（ビューの移動）	白い境界線を持つ移動可能で透明な長方形を表示します。クリックすると PLUS+1 GUIDE ウィンドウの図面エリアがこの長方形の境界に入るように中心が移動します。
Zoom（ズーム）	**In**： ズームインして、ポインタをクリックした場所を **PLUS+1 GUIDE** ウィンドウの図面エリアの中央に移動します。 **Out**：ズームアウトして、ポインタをクリックした場所を **PLUS+1 GUIDE** ウィンドウの図面エリアの中央に移動します。 **Area**： **PLUS+1 GUIDE** ウィンドウの図面エリアを 2 箇所クリックしたポイントで決まる範囲にズームします。 **Fit Page**： ページ内の全てのアイテムが **PLUS+1 GUIDE** ウィンドウの図面エリア内にフィットするようにビューのサイズを変更します。 **100%**： **PLUS+1 GUIDE** ウィンドウの図面エリアを最小ズームビューと最大ズームビューの中間サイズにズームします。
Module Viewer	**Module Viewer** ウィンドウを表示します。 **Module Viewer** ウィンドウはメインの PLUS+1 GUIDE ウィンドウとは独立したウィンドウで、SCS ファイルを編集することが出来る追加ウィンドウです。
Manager(マネージャ)	**Project Manager** タブと **Page Navigator** タブのある固定 **Manager** ペインの表示／非表示を切り換えます。 **Project Manager** タブ：アプリケーションの作成とコンパイルのためにインストールしたハードウェア記述のツリービューを表示します。 **Hardware** タブのハードウェア記述を選択してからこのタブにドラッグします。

PLUS+1 GUIDE ウィンドウメニュー

項目	説明
Page Navigator(ページナビゲータ)	アプリケーションモジュールのページのツリービューを含みます。ビュー内のページ名をクリックして、図面エリアの選択ページを表示します。
Inspector(インスペクタ)	固定された **Inspector** タブの表示／非表示を切り換えます。 このタブを使用して、**Query/Change** ツールで選択したアイテムのプロパティを確認し、変更します。
Compiler Messages(コンパイラメッセージ)	**Compiler Messages** タブ、**Error Messages** タブ、**Build Messages** タブのある固定ペインの表示／非表示を切り換えます。これらのタブはコンパイラメッセージを表示します。 **Compiler Messages** タブ: プロジェクトフォルダにある ***Screen.tmp*** ファイルに書き出すコンパイラメッセージを表示します。 **Error Messages** タブ: コンパイラエラーメッセージを表示します。エラーはコンパイルプロセスを中止します。 **Build Messages** タブ: ダウンロード形式のファイルにアプリケーションをコンパイルするときにコンパイラからのメッセージを表示します。
Selector(セレクタ)	**Hardware** タブ、**Preview** ペインを含む **Component** タブ、**Function** タブのある固定 **Selector** ペインの表示／非表示を切り換えます。 **Hardware** タブ: 特定 PLUS+1 モデル用ハードウェア記述のツリービューを表示します。 各ハードウェア記述は、特定 PLUS+1 ハードウェアモデル用のダウンロード可能なファイルを作成・コンパイルするのに必要なリソースにリンクしています。ここからハードウェア記述を **Project Manager** タブにドラッグします。 **Component** タブ: アプリケーションを構築するのに使用できるコンポーネントのツリービューを表示します。コンポーネントを図面エリアにドラッグしてください。 **Preview** ペインは、選択したコンポーネントのプレビューを示します。 **Function** タブ: アプリケーションで使用できる高度な機能のツリービューを表示します。機能を図面エリアにドラッグします。
Default Layout(デフォルトレイアウト)	デフォルトレイアウト表示／非表示を切り換えます。 デフォルトレイアウトでは、**PLUS+1 GUIDE** ウインドウが左側の **Manager** ペインと **Inspector** ペイン、下部の **Compiler** ペイン、右側の **Selector** ペインを固定して表示します。
Toolbars(ツールバー)	**File**、**Edit**、**View**、**Compile**、**Add**、**Help** といった各機能のボタン設定表示／非表示を切り換えます。
Compile(コンパイル)	このメニューのコマンドは、コンパイル機能を制御します。
Compile All(全コンパイル)	全アプリケーションモジュールをコンパイルして、ダウンロード形式のファイルを作ります。
Compile Changed(変更コンパイル)	最近の **Compile All** 以降に変更したモジュールのみをコンパイルします。 複数のモジュールがあるときは、このコマンドを使用して、コンパイル時間を節約します。
Error Check All(全エラーチェック)	コンパイルして、全モジュールのエラーをチェックします。 **Error Check All**、**Error Check Changed** のコマンドは、ダウンロード可能なファイルを生成しません。
Error Check Changed(変更エラーチェック)	コンパイルして、最近の **Compile All** 以降に変更したモジュール内のエラーのみをチェックします。
Stop Compile (コンパイル停止)	全コンパイルプロセスを停止します。
Setup(セットアップ)	PLUS+1 GUIDE のプログラミング環境をセットアップします。
Toggle Display Grid(表示グリッドのトグル)	**Toggle Display Grid:** ディスプレイグリッドを On/Off します。 ディスプレイグリッドとは **PLUS+1 GUIDE**、**Page Interface Editor** および **Module Viewer** ウィンドウの図面エリアで表示されるグリッドです。

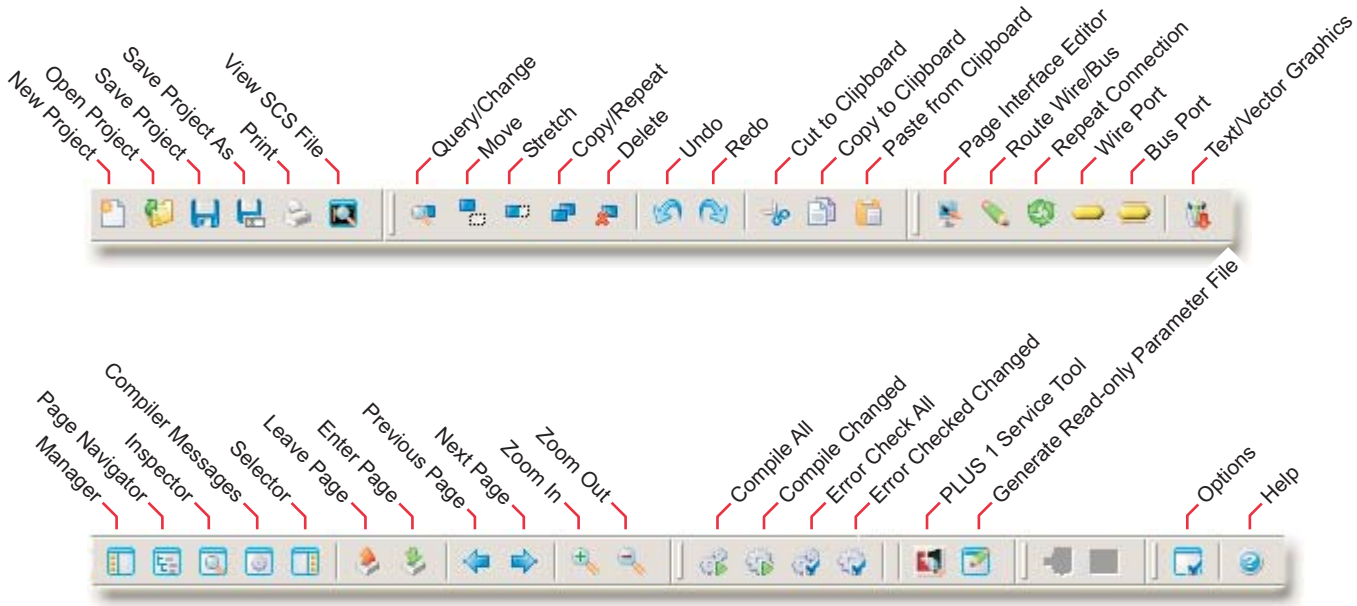
PLUS+1 GUIDE ウィンドウメニュー

項目	説明
Layers（階層）	**Visible Layers**（可視階層）: **Define Visible/Define Invisible Objects/Layers**（オブジェクト／階層の可視／不可視の定義）ウインドウを表示します。 PLUS+1 GUIDE アプリケーションが図面エリア内で見せたり隠したりするコンポーネントやルートのようなオブジェクトのクラスを、このウインドウを使用して選択します。 **Active Layers**（アクティブ階層）: この PLUS+1 GUIDE リリース版はこの機能に対応していません。
Display（ディスプレイ）	**Graphics and Text**（グラフィックスとテキスト）: **Define Drawing Mode Filled Area & Text**（図面のモード、塗りつぶしエリア、テキストの定義）ウインドウを表示します。 このウインドウを使用して PLUS+1 GUIDE ウィンドウの図面エリア内のテキストやラインの表示特性を変更します。変更は全てのプロジェクトに適用されます。 PLUS+1 GUIDE のデフォルト設定値は、全ラインについては **True Size**、テキストについては **Center Line** です。 **Drawing Area Pointer**（図面エリアポインタ）: **Graphics Pointer Setup**（グラフィックスポインタセットアップ）ウインドウを表示します。 このウインドウを使用して、ポインタ表示、ポインタ解像度の変更および、x軸と y 軸の0基準の設定をします。 **Color Schemes**（カラースキーム）: **Classic** スキームを選択すると PLUS+1 GUIDE ウィンドウの図面エリアの背景を黒にします。**White** スキームを選択すると図面エリアの背景をオフホワイトに設定します。
Install Hardware（ハードウェアのインストール）	**Install Hardware**（ハードウェアのインストール）ウインドウを表示します。 このウインドウを使用して、ハードウェア定義をブラウズして **Hardware** タブに追加します。
Install Function Library（インストール機能ライブラリ）	**Install Function Library**（機能ライブラリのインストール）ウインドウを表示します。 このウインドウを使用して、ライブラリファイルをブラウズして **Library** タブに追加します。
Uninstall Hardware（ハードウェアのアンインストール）	**Uninstall**（アンインストール）ウインドウを表示します。 このウインドウを使用して、ハードウェア定義を **Hardware** タブから削除します。
Uninstall Function Library（機能ライブラリのアンインストール）	**Uninstall Function Library**（機能ライブラリのアンインストール）ウインドウを表示します。 このウインドウを使用して **Library** タブから機能ライブラリをブラウズして削除します。
Install Font	**Install Font** ウィンドウを表示します。 このウィンドウを使用して PLUS+1 グラフィックディスプレイのプロジェクトで使うことのできる追加フォントを作成します。 詳細については xxx ページの xxx を参照してください。
Uninstall Font	**Uninstall Font** ウィンドウを表示します。 このウィンドウを使用してインストール済みの全てのフォントからアンインストールするフォントを選択します。
Tools Palette（ツールパレット）	**Icon Menu**（アイコンメニュー）ウインドウを表示します。
Options（オプション）	**Options**（オプション）ウインドウを表示します。 このウインドウを使用して、**Undo/Redo** コマンドの有効化などの PLUS+1 GUIDE オプションを設定します。
Add（追加）	このメニューのコマンドは、ページアウトライン、バス、ポート、テキスト、その他のエレメントをアプリケーションに追加します。
Page（ページ）	PLUS+1 GUIDE ウィンドウの図面エリア内でクリック＆ドラッグした部分にページを追加します。 追加したページのトップビューを編集するには、ページに入ってから **Page Interface Editor** をクリックします。 詳細な情報については *82* ページの*ページインタフェースエディタウィンドウ*を参照してください。
Page Interface Editor（ページインターフェイスエディタ）	作業中のページのトップビューページを **Page Interface Editor** ウィンドウに表示します。 **Page Interface Editor** ウィンドウを使用してページのトップビューを編集します。 詳細な情報については *82* ページの*ページインタフェースエディタウィンドウ*を参照してください。
Wire Port（ワイヤポート）	図面エリアのクリックした位置に単一信号ポートを追加します。
Bus Port（バスポート）	図面エリアのクリックした位置にマルチ信号ポートを追加します。

PLUS+1 GUIDE ウィンドウメニュー

項目	説明
Route Wire/Bus（ルートワイヤ／バス）	PLUS+1 GUIDE ウィンドウの図面エリアのクリックした位置から緑色の単一信号ワイヤまたは赤色のマルチ信号バスのいずれかの配線を始めます。 バスポートから配線を始めるとバスになります。ワイヤポートから配線を始めるとワイヤになります。 K でワイヤとバスのルートをトグルで切り換えます。 F9 で接続されていないルートを終端します。
Repeat Connection（接続の繰り返し）	**PLUS+1 GUIDE** ウィンドウの図面エリア内で、選択したルート接続を繰り返します。 繰り返し作成された接続は自動的に元の名前に数字が追加されます。***Signal_Name***、***Signal_Name2***、***Signal_Name3***、***Signal_Name4*** など。
Text/Vector Graphics（テキスト／ベクトルグラフィックス）	テキストとグラフィック用コマンドのメニューを表示します。 このコマンドを使用してテキストやグラフィックエレメントを **PLUS+1 GUIDE** ウィンドウの図面エリアに追加します。
Tools（ツール）	このメニューのコマンドは、ダウンロードしたサービスアプリケーションへのツールのオープン、モジュールのエラーの修復、PLUS+1 GUIDE ライセンスの管理をします。
PLUS+1 Service Tool（PLUS+1 サービスツール）	PLUS+1 サービスツールプログラムを開きます。 このツールを使用して、アプリケーションを PLUS+1 ハードウェアにダウンロードしたり、アプリケーション値をモニター・変更したりします。 このサービスプログラムは PLUS+1 GUIDE プログラムとは別個に開くことができます。
Optimize Nets（ネットの最適化）	ブロックまたはページをインポートしたときに壊れたルートを修復するのに使用します。 メモリから不要なデータを取り除き、ツール性能を向上させ、アプリケーションファイルを小さくするのに使用します。
View Logical Net（論理ネットの表示）	**Logical Net**（論理ネット）ウインドウを表示します。 このウインドウを使用して、単一ページや複数ページにわたって、選択した論理ネットの接続をトレースします。
View Net Names（ネット名の表示）	浮いているエントリー、インターフェイスポートのミスマッチ、ネットエラーを示す様々なウインドウを表示します。 このリストのアイテムをクリックして、問題のあるページのプレビューを見ます。
License Manager（ライセンスマネージャ）	**Select License**（ライセンス選択）ウインドウを表示します。 このウインドウを使用して、PLUS+1 GUIDE ライセンスを管理します。 **user ID** は、ご利用のコンピュータ固有のものです。この番号をザウアーダンフォス社に送り、ライセンス番号を入手し、PLUS+1 プログラムのロックを解除します。
Help（ヘルプ）	このメニューのコマンドは、PLUS+1 GUIDE プログラムについてのヘルプと情報を表示します。
Help Contents（ヘルプコンテンツ）	ユーザーマニュアルのハイパーリンク付きオンラインヘルプ版を開きます。
User Manual (ユーザーマニュアル)	印刷用にフォーマットされたユーザーマニュアル PDF 版を開きます。
Release Notes（リリースノート）	PLUS+1 GUIDE プログラム用 Readme Help ファイルを表示します。
Create Troubleshooting File（トラブルシューティングファイルの作成）	**Create Troubleshooting File**（トラブルシューティングファイルの作成）ウインドウを表示します。 このウインドウを使用して、ザウアーダンフォス社のヘルプデスクに電子メールで送るための圧縮フォルダを作成します。このフォルダには、プロジェクトファイルおよびその他の役に立つトラブルシューティング情報が入っています。
About PLUS+1 GUIDE（PLUS+1 GUIDE について）	About PLUS+1 GUIDE（PLUS+1 GUIDE について）ウインドウを表示します。 このウインドウを使用して、PLUS+1 GUIDE のバージョンとライセンス情報を確認します。

PLUS+1 GUIDE ウィンドウツールバー



PLUS+1 GUIDE ウィンドウのツールバーのボタンは、よく使用する PLUS+1 GUIDE コマンドにアクセスします。

スクリーンエディタは、上図のグレイアウトされた（使用できない）ボタンを使います。

メニューバーでアクセスする全コマンドのリストは***46*** ページの ***PLUS+1 GUIDE****ウィンドウメニュー*を参照してください。

ツールバー

ボタン		説明
	New Project （新規プロジェクト）	**Create New Project** ウインドウを表示します。 このウインドウを使用して、新しいプロジェクトに名前を付け、プロジェクトファイル用のフォルダを作成します。
	Open Project （プロジェクトを開く）	**Open Project** ウインドウを表示します。 このウインドウを使用して、*P1P* および *P1X* プロジェクトファイルを見つけて開きます。
	Save Project （プロジェクトの保存）	1 つの P1P ファイルに圧縮せずに現在のプロジェクトファイルを保存します。
	Save Project As （名前を付けてプロジェクトを保存）	**Save Project As** ウインドウを表示します。 このウインドウを使用して、新しいフォルダ内に新しい名前で現在のプロジェクトを保存します。
	Print （印刷）	**Print**（印刷）ウインドウを表示します。 このウインドウを使用してアプリケーションからページをプリントアウトします。
	Module Viewer	**Module Viewer** ウィンドウを表示します。 **Module Viewer** ウィンドウは PLUS+1 GUIDE のメインウィンドウとは独立した追加ウィンドウを表示し、SCS ファイルを編集することができます。

ツールバー

ボタン		説明
	Query/Change （クエリー／変更）	**PLUS+1 GUIDE** ウィンドウの図面エリア内のアイテムのプロパティ（データ型、文字、機能など）を変更するときに使用します。 プロパティを変更したいアイテムをクリックしてください。選択したアイテムに適したダイアログボックスが表示されます。このウィンドウ内でプロパティを変更してください。
	Move（移動）	**PLUS+1 GUIDE** ウィンドウの図面エリア内で、選択したアイテムを移動します。
	Stretch （ストレッチ）	選択したルートのセグメントをストレッチし、選択したアイテムを移動します。 ルートをクリックしてルートに頂点を追加します。新しい頂点はルート上の他の頂点の位置を変えずに移動させることができます。
	Copy/Repeat （コピー／リピート）	**PLUS+1 GUIDE** ウィンドウの図面エリア内で、選択したアイテムをコピーします。選択されたアイテムは白くなります。クリックするとコピーしたアイテムの位置を確定します。Esc を押してコピーしたアイテムをキャンセルします。 このコマンドは現在の図面エリア内でのみ有効です。 リンク機能によりコピーしたページを自動的にリンクします。詳細については *472* ページの*リンクページについて*を参照してください。
	Delete （削除）	**PLUS+1 GUIDE** ウィンドウの図面エリアで、選択したアイテムを削除します。選択したアイテムは白くなります。 独立したアイテムや同一アイテムを選択すると、**Attributes**（属性）ウインドウが表示されます。**OK** をクリックして選択アイテムを削除します。 **Select Item Class** ウインドウは、複数の異なるアイテムを選択するときに表示されます。このウインドウを使用して、削除したいアイテムを選択します。 星印(*)で削除されるアイテムを指定します。 ダッシュ(-)で削除しないアイテムを指定します。 **Select Item Class** ウィンドウで **OK** をクリックして選択アイテムを削除します。
	Undo/Redo （アンドゥ／リドゥ）	**Undo:** プログラム操作を取り消します。 **Redo**: **Undo** 操作を取り消します。 コンピュータのメモリによって、アンドゥーできる回数が決まります。 **Option** ウインドウで Undo/Redo 機能の切り替えが可能です。このウインドウを表示させるには、ツールバーで Options をクリックします。
	Page Interface Editor （ページインタフェースエディタ）	**Page Interface Editor** ウィンドウ内に作業中のページのトップビューページを表示します。 **Page Interface Editor** ウィンドウを使用して、ページのトップビューを編集します。 詳細については、*82* ページの*ページインタフェースエディタウィンドウ*を参照してください。
	Cut to Clipboard （クリップボードに切り取り）	**PLUS+1 GUIDE** ウィンドウの図面エリア内で、選択したアイテムを削除してクリップボードにコピーします。
	Copy to Clipboard （クリップボードにコピー）	**PLUS+1 GUIDE** ウィンドウの図面エリア内で、選択したアイテムをクリップボードにコピーします。
	Paste from Clipboard （クリップボードから貼り付け）	**PLUS+1 GUIDE** ウィンドウの図面エリア内で、クリップボードの内容を貼り付けます。 リンク機能により、クリップボードからコピーされた複製ページの内容は自動的にリンクされます。詳細については *472* ページ、*リンクページについて* を参照してください。

ツールバー

ボタン		説明
	Route Wire/Bus	PLUS+1 GUIDE ウィンドウの図面エリアのクリックした位置から緑色の単一信号ワイヤまたは赤色のマルチ信号バスのいずれかの配線を始めます。 バスポートから配線を始めるとバスになります。ワイヤポートから配線を始めるとワイヤになります。 K でワイヤとバスのルートをトグルで切り換えます。 F9 で接続されていないルートを終端します。
	Repeat Connection	**PLUS+1 GUIDE** ウィンドウの図面エリア内で、選択したルート接続を複製します。 繰り返された接続は自動的に元の名前に番号が追加されます。*Signal_Name*、*Signal_Name2*、*Signal_Name3*、*Signal_Name4* などです。
	Wire Port	図面エリアのクリックした位置に単一信号ポートを追加します。
	Bus Port	図面エリアのクリックした位置にマルチ信号ポートを追加します。
	Text/Vector Graphics	テキストとグラフィック用コマンドのメニューを表示します。 このコマンドを使用してテキストやグラフィックエレメントを **PLUS+1 GUIDE** ウィンドウの図面エリアに追加します。
	Manager	Manager、Page Navigator、Inspector、Compiler Messages、Selector の各ボタンは、デフォルトレイアウトを選択したときに固定タブの表示を制御します。デフォルトレイアウトを選択するには、**View** メニューで **Default Layout** をクリックします。 このボタンは、**Project Manager** タブと **Page Navigator** タブを含む固定 **Manager** ペインの表示／非表示を切り換えます。 **Project Manager** タブ：アプリケーションの作成とコンパイルのためにインストールしたハードウェア記述のツリービューを表示します。 **Hardware** タブのハードウェア記述を選択してからこのタブにドラッグします。
	Page Navigator	アプリケーションのページのツリービューを含む **Page Navigator** タブを表示します。ビュー内のページ名をクリックして、図面エリアの選択ページを表示します。
	Inspector	固定 **Inspector** タブの表示／非表示を切り換えます。 このタブを使用して、Query/Change ボタンで選択したアイテムのプロパティを確認して変更します。
	Compiler Messages	**Compiler Messages** タブ、**Error Messages** タブ、**Build Messages** タブを含む固定ペインの表示／非表示を切り換えます。これらのタブはコンパイラメッセージを表示します。 **Compiler Messages** タブ：プロジェクトフォルダにある Screen.tmp ファイルに書き出すコンパイラメッセージを表示します。 **Error Messages** タブ：コンパイラエラーメッセージを表示します。エラーはコンパイルプロセスを中止します。 **Build Messages** タブ： ダウンロード可能形式のファイルにアプリケーションをコンパイルするときにコンパイラからのメッセージを表示します。
	Selector	**Hardware** タブ、**Preview** を含む **Component** タブ、**Function** タブのある固定セレクタペインの表示／非表示を切り換えます。 **Hardware** タブ：特定の PLUS+1 モデル用ハードウェア記述のツリービューを表示します。 各ハードウェア記述は、特定の PLUS+1 ハードウェアモデル用のアプリケーションを作成・コンパイルするのに必要なリソースにリンクしています。ここからハードウェア記述を **Project Manager** タブにドラッグします。 **Component** タブ：アプリケーションを構築するのに使用できるコンポーネントのツリービューを表示します。必要なコンポーネントを図面エリアにドラッグします。 **Preview** ペインは、選択したコンポーネントのプレビューを示します。 **Function** タブ：アプリケーションで使用できる高レベルファンクションのツリービューを表示します。必要な機能を図面エリアにドラッグします。

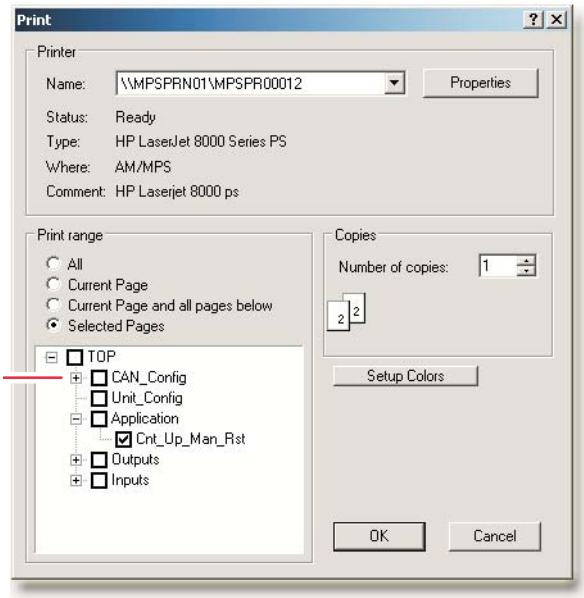
ツールバー

ボタン		説明
	Leave Page	現在のページから出ます。
	Enter Page	選択したページに入ります。 ページに入るには、ページ境界の内側をクリックするか、ページのポートをドラッグします。
	Previous Page	ページナビゲーションの履歴から 1 ページ前に戻ります。
	Next Page	ページナビゲーションの履歴から 1 ページ次に進みます。
	Zoom In	ズームインして、ポインタをクリックした場所を **PLUS+1 GUIDE** ウィンドウの図面エリアの中央に移動します。
	Zoom Out	ズームアウトして、ポインタをクリックした場所を **PLUS+1 GUIDE** ウィンドウの図面エリアの中央に移動します。
	Compile All	全アプリケーションモジュールをコンパイルして、ダウンロード形式のファイルを作ります。
	Compile Changed	最近の **Compile All** 以降に変更したモジュールのみをコンパイルします。 複数のモジュールがあるときは、このコマンドを使用して、コンパイル時間を節約します。
	Error Check All	全モジュールのエラーをチェックします。 **Error Check All**、**Error Check Changed** のコマンドは、ダウンロード可能なファイルを生成しません。
	Error Check Changed	コンパイルして、最近の **Compile All** 以降に変更したモジュール内のエラーのみをチェックします。
	PLUS+1 Service Tool	PLUS+1 サービスツールアプリケーションを開きます。 このツールを使用して、アプリケーションを PLUS+1 ハードウェアにダウンロードしたり、アプリケーション値をモニター・変更したりします。 このサービスアプリケーションは PLUS+1 GUIDE アプリケーションとは別個に開くことができます。
	Generate Read-only Parameter File	**Generate Read-only Parameter File** ウィンドウを表示します。 このウィンドウを用いてアプリケーションパラメータを含む lhx ファイルを生成します。詳細な情報は *443* ページの *読み込み専用パラメータファイルの作成* を参照してください。
	Options	**Options** ウィンドウを表示します。 このウィンドウを用いて **PLUS+1 GUIDE** のオプションを設定します。
	Help	***PLUS+1 GUIDE User Manual*** を PDF 形式で表示します。

Print ウィンドウ

ツールバー >

または **File** メニュー> **Print**



このウインドウを使用して PLUS+1 GUIDE アプリケーションからページをプリントアウトします。

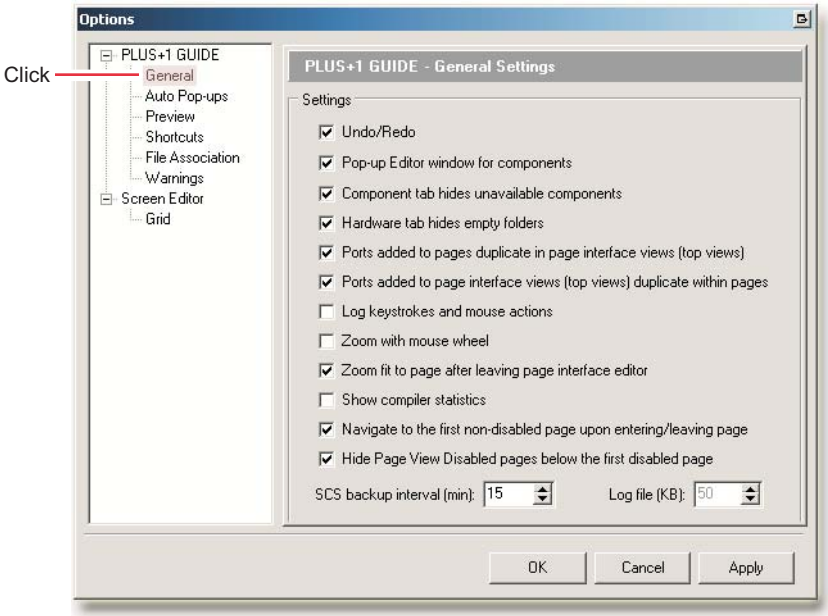
Print ウィンドウ

項目	説明
Name	プリンタを選択します
Properties	**Printer Properties** ウィンドウを表示します。 このウィンドウを用いて選択したプリンタに固有の機能を有効化します。各プリンタモデルに対して異なる **Printer Properties** ウィンドウがあります。
Copies	各ページに対して印刷する部数を設定します。
Print range	**All**—アプリケーションの全てのページを印刷します。 **Current page**—PLUS+1 GUIDE ウィンドウの図面エリア内に現在表示されているページを印刷します。 **Current page and all pages below**—現在のページと、現在のページ内の全ての（サブ）ページを印刷します。 **Selected pages**—Print Selection Tree でチェックを入れたページを印刷します。
Print Selection Tree	アプリケーション内の全てのページのツリービューを表示します。 印刷するページを選択するために、**Pages** オプションを選択して印刷したいページをチェックします。
Setup Colors	**Assign Pens** ウィンドウを表示します。 このウィンドウを用いて、図面エリア内のアイテムにデフォルトで割り当てられている色を再マップした配色で印刷します。

Options ウィンドウ—General Settings（一般設定）

ツールバー >

または **Setup** メニュー> **Options** > **Options** ウィンドウ



- **General Settings**（一般設定）で PLUS+1 GUIDE の一般オプションを設定します。
- **Auto Pop-ups Settings** でポップアップオプション（*58*ページ参照）を設定します。
- **Preview Settings**（*61*ページ参照）で**Component** タブや他のタブ内でのPreviewペインを設定します。
- **Shortcuts Settings**（*62*ページ参照）で**Classic**, **Windows**, あるいは**Custom** のショートカットキーを設定します。
- **Grid Settings**（*66*ページ参照）でScreen Editor内で表示されるグリッドの設定をします。

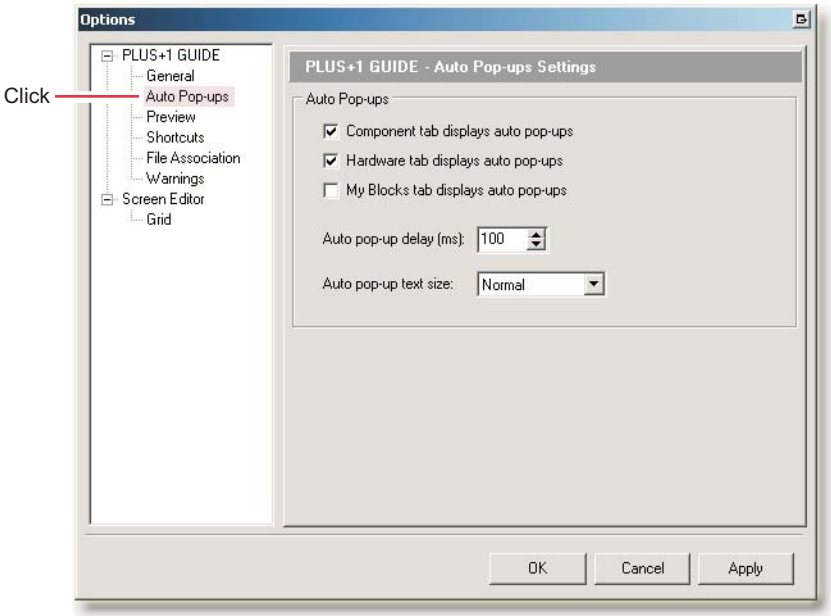
Options ウィンドウ　General Settings

Item	Description
Undo/Redo（アンドゥ／リドゥ）	チェックするとアンドゥ／リドゥのコマンドを有効化します。
Pop-up Editor window for components	チェックすると Query/Change ツールでコンポーネントをクリックしたときの Edit Value ウィンドウのポップアップを有効化します。
Component tab hides unavailable components	チェックすると **Component** タブに選択したハードウェアで使用できないコンポーネントが表示されなくなります。
Hardware tab hides empty folders	チェックすると **Hardware** タブにハードウェアツリーの空フォルダが表示されなくなります。
Ports added to pages duplicate in page interface views (top views)	チェックすると、ページインタフェースビュー（トップビュー）に追加したポートがそのページ内にも追加されます。 **Page Interface Editor** ウィンドウで追加したポートはページ内の左上隅に現れます。

Options ウィンドウ General Settings

Item	Description
Ports added to page interface views (top views) duplicate within pages	チェックすると、ページに追加されたポートがそのページのページインタフェースビュー（トップビュー）にも追加されます。 ページに追加されたポートは **Page Interface Editor** ウィンドウの上左隅に現れます。
Log keystrokes and mouse actions	チェックすると ***_Plus1GUIDE.log*** ファイルを作成します。このファイルはプロジェクトフォルダに作成され、マウス操作とキー操作を記録し PLUS+1 GUIDE アプリケーションの問題点を診断するのに役立ちます。
Zoom with mouse wheel	チェックすると、Drawing Area をマウスホイールでズームすることができるようになります。
Zoom fit to page after leaving Page Interface Editor	チェックすると、**Page Interface Editor** ウィンドウを閉じると自動的にビューをホームに戻します。（**Page Interface Editor** を閉じた後 Home を押すのと同じ動作）
Show compiler statistics	チェックすると、コンパイルが正常に実行された場合に **Compile Progress** ウィンドウで **Total pages count** と **View disabled pages count** を表示します。これらの情報は **Compile Progress** ウィンドウの **User Message** ペインに表示されます。
Navigate to the first non-disabled page upon entering/leaving page	チェックすると、**Page View Access** プロパティが Disabled に設定されているページに入ったり出たりする場合、**Page View Access** プロパティが **Normal, Force Enabled** あるいは **Read-only** と設定されている最初のページにジャンプするようになります。
Hide Page View Disable pages below the first disabled page	チェックすると、**Page Navigator** タブで最初のページとすぐ下のページの両方のページの **Page View Access** のプロパティが同じ **Disabled** に設定されている場合、すぐ下のページを表示しません。 チェックを外すと、**Page Navigator** タブに全てのページを表示します。 **Page Navigator** では **Page View Access** プロパティが **Disabled** のページはグレー表示になります。
SCS backup interval	PLUS+1 GUIDE プログラムが自動的に SCS ファイルのバックアップを取る頻度を設定します。図面エリアで変更が行われると次のバックアップまでのカウントダウンが開始します。 PLUS+1 GUIDE プログラムはバックアップファイル（~***.scs）を現在のプロジェクトフォルダ内に作成します。 **範囲: 1～999 分**
Log file	***_Plus1GUIDE.log*** ファイルのサイズを定義します。

Options ウィンドウ—Auto Pop-ups Settings（自動ポップアップ設定）

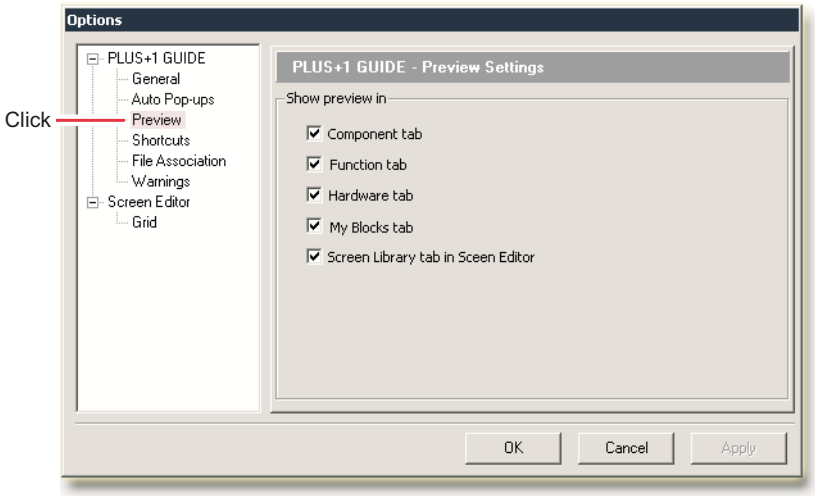


Auto Pop-ups Settings は自動ポップアップのオプションを設定します。

Options ウィンドウ　Auto Pop-ups Settings

Item	Description
Component tab displays auto poups	チェックすると、**Component** タブ内で展開されていないアイテム上にマウスを置いたとき、そのツリー内にある全てのアイテムをポップアップメニューで表示します。
Hardware tab displays auto pop-ups	チェックすると、**Hardware** タブ内で展開されていないアイテム上にマウスを置いたとき、そのツリー内にある全てのアイテムをポップアップメニューで表示します。
My Blocks tab displays auto pop-ups	チェックすると、**My Blocks** タブ内で展開されていないアイテム上にマウスを置いたとき、そのツリー内にある全てのアイテムをポップアップメニューで表示します。
Auto pop-up delay (ms)	ポップアップメニューが表示されるまでの遅延時間を設定します。 **範囲**: **10–5000 ms**
Auto pop-up text size	– **Normal**—ポップアップメニューがアイコン付きで表示されます – **Small (No Icons)**—ポップアップメニューがアイコンなしの小さい文字で表示されます

Options ウィンドウ—Preview Settings（プレビュー設定）

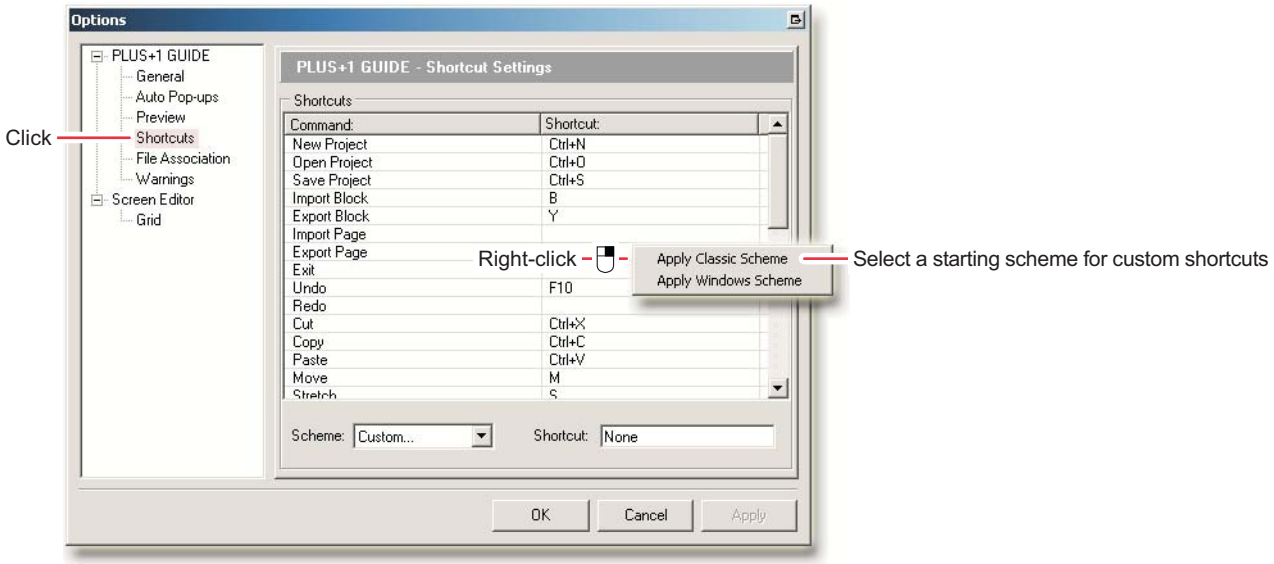


Preview Settings で **Component** タブや他のタブ内での **Preview** を有効化します。

Options ウィンドウ　Preview Settings

項目	説明
Component tab （コンポーネントタブ）	チェックを入れると、**Component** タブの **Preview** が使えます。 このタブでクリックしたコンポーネントを **Preview** パネルでプレビューします。
Function tab （ファンクションタブ）	チェックを入れると、**Function** タブの **Preview** が使えます。 このタブでクリックしたファンクションを **Preview** パネルでプレビューします。
Hardware tab （ハードウェアタブ）	チェックを入れると、**Hardware** タブの **Preview** が使えます。 このタブでクリックしたハードウェアを **Preview** パネルでプレビューします。
My Blocks tab （マイブロックタブ）	チェックを入れると、**My Blocks** タブの **Preview** が使えます。 このタブでクリックしたブロックを **Preview** パネルでプレビューします。
Screen Library tab in Screen Editor （スクリーンエディタのスクリーンライブラリタブ）	チェックを入れると、**Screen Library** タブの **Preview** が使えます。 このタブでクリックしたアイテムを **Preview** パネルでプレビューします。

Options ウィンドウ—Shortcuts Settings（ショートカットの設定）



Shortcuts Settings enable **Classic**, **Windows**, or **Custom** shortcut key schemes.

Options ウィンドウ　Shortcuts Settings

項目	説明
Shortcuts	**Classic**, **Windows**, または **Custom** のショートカット体系の **Command** と **Shortcut** キーの組み合わせを表示します。 Classic と Windows ショートカット体系は変更できません。 Custom 体系を編集するには、Scheme で Custom を選択します。Shortcuts ペインで右クリックしポップアップメニューを表示させます。このメニューで Apply Classic Scheme または Apply Windows Scheme のどちらかをクリックし、カスタムショートカットの基準のショートカット体系を選択します。Command をクリックし、Shortcut フィールドで所望のショートカットを入力します。
Scheme	**Classic**, **Windows**, あるいは **Custom** のショートカットキーの組み合わせを選択します。 **Custom** を選択して独自のショートカットキーの組み合わせを作成します。
Shortcut	**Scheme** で **Custom** を選択するとこのフィールドを編集できます。 選択した **Command** に対してショートカットを入力します。

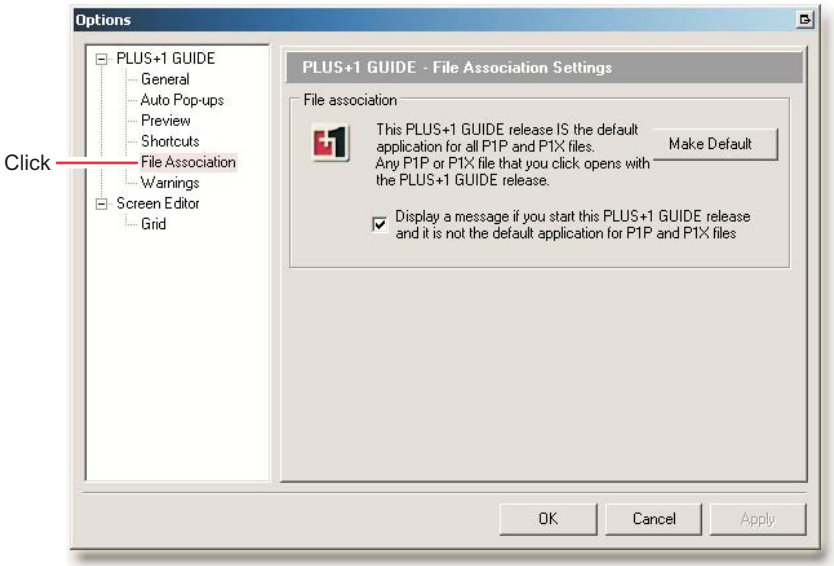
Classic および Windows ショートカットキー体系

コマンド	Classic ショートカット	Windows ショートカット	Comment
New Project	Ctrl+N	Ctrl+N	
Open Project	Ctrl+O	Ctrl+O	
Save Project	Ctrl+S	Ctrl+S	
Import Block	B	B	
Export Block	Y	Y	
Import Page			
Export Page			
Exit			

Classic および Windows ショートカットキー体系

コマンド	Classic ショートカット	Windows ショートカット	Comment
Undo	F10	Ctrl+Z	
Redo		Ctrl+Y	
Cut	Ctrl+X	Ctrl+X	
Copy	Ctrl+C	Ctrl+C	
Paste	Ctrl+V	Ctrl+V	
Move	M	M	
Stretch	S	S	
Delete	X	Del	
Copy/Repeat	C	C	
Query/Change	Q	Q	
Highlight Connections			
Remove Highlighting			
Enter Page	E	E	
Leave Page	L	L	
Refresh	F4	F4	
Set Center	Ins	Ins	
Move View	Ctrl+F3	Ctrl+F3	
Zoom In	PgUp	PgUp	
Zoom Out	PgDn	PgDn	
Zoom Area	F3	F3	
Zoom Fit Page	Home	Home	
Toggle Display Grid	G	G	
Route Wire/Bus	R	R	
Add Text			
View Logical Net	V	V	
Help	F1	F1	カスタマイズ不可
Optimize Nets			
Transfer between Pages			

Options ウィンドウ—File Association Settings（ファイル関連付け設定）



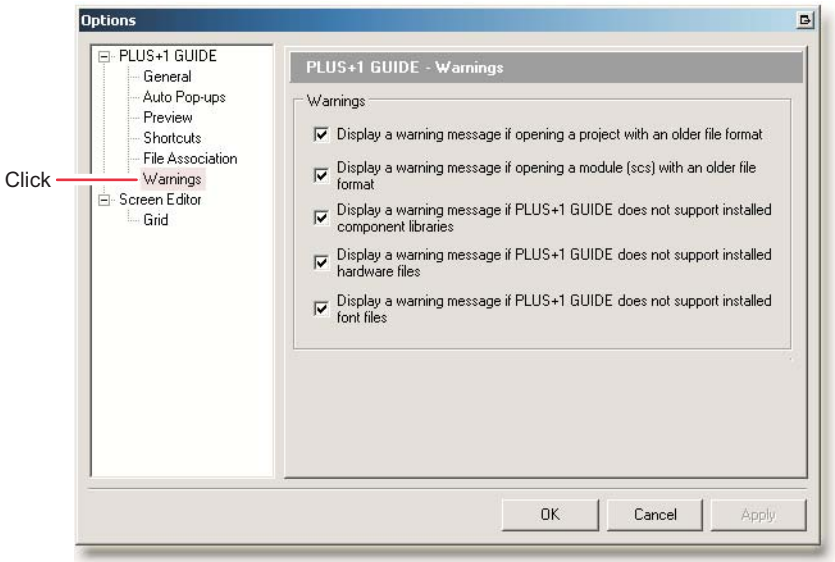
File Association Setting は異なるリリースの PLUS+1 GUIDE を一つの PC 上で実行するときに役立ちます。

- 同一 PC 上にインストールされた異なる PLUS+1 GUIDE リリースの実行はリリース 3.3 以降からサポートされます。
- 同時に実行できる PLUS+1 GUIDE プログラムは一つだけです。

Options ウィンドウ File Association Settings

Item	Description
Make Default	チェックするとこの PLUS+1 GUIDE リリースを P1X と P1P ファイルのデフォルトアプリケーションとして設定します。 P1X あるいは P1P 拡張子の付いたファイルをクリックすると自動的にこのリリースの PLUS+1 GUIDE で Open します。
Check at startup	チェックすると、この PLUS+1 GUIDE リリースが P1X と P1P のデフォルトアプリケーションでないときメッセージを表示します。

Options ウィンドウ—Warnings（警告表示）



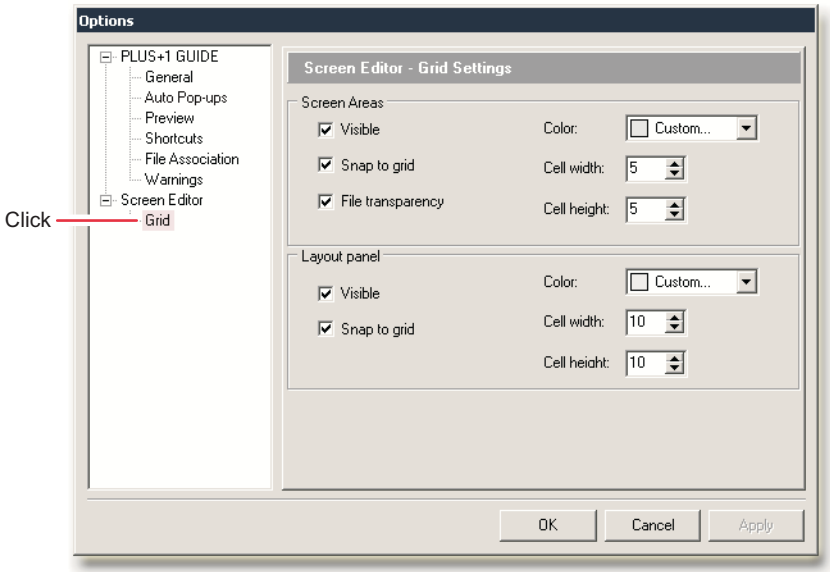
Warnings は同じ PC 上で実行される異なるリリースの PLUS+1 GUIDE に関する警告メッセージを管理します。

- 同一 PC 上にインストールされた異なる PLUS+1 GUIDE リリースの実行はリリース 3.3 以降からサポートされます。
- 同時に実行できる PLUS+1 GUIDE プログラムは一つだけです。

Options ウィンドウ　Warnings

項目	説明
Project—older file format	チェックすると、現在の PLUS+1 GUIDE リリースが古いファイルフォーマットのプロジェクトを開こうとするときに警告メッセージを表示します。
Module—older file format	チェックすると、現在の PLUS+1 GUIDE リリースが古いファイルフォーマットのモジュール（SCS ファイル）を開こうとするときに警告メッセージを表示します。
Unsupported component libraries	チェックすると、現在の PLUS+1 GUIDE リリースがインストール済みの Component ライブラリをサポートしていないときに警告メッセージを表示します。
Unsupported hardware files	チェックすると、現在の PLUS+1 GUIDE リリースがインストール済みの Hardware ファイルをサポートしていないときに警告メッセージを表示します。
Unsupported font files	チェックすると、現在の PLUS+1 GUIDE リリースがインストール済みのフォントファイルをサポートしていないときに警告メッセージを表示します。

Options ウィンドウ—Grid Settings



Grid Settings で Screen Editor で使用する以下の二つのグリッドについて設定します。

- **Layout pane** グリッド—このグリッドはレイアウトペインの全体に設定されます。このグリッドは Define Area ページ内で Screen Area をレイアウトするときに使います。
- **Screen area** グリッド—このグリッドは各 Screen Area 内に表示されます。このグリッドは **Define Screen** ページ内で Screen Library アイテムをレイアウトするときに使います。

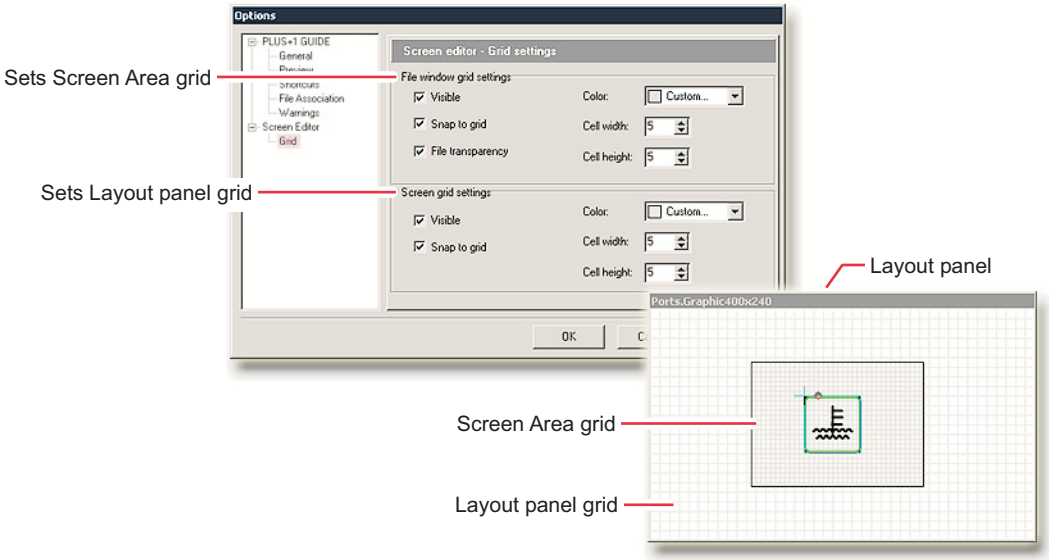
Options ウィンドウ　Grid Settings

Item	Description
Screen Areas（スクリーンエリア）	各スクリーンエリアに表示されるグリッドの特徴を設定します。 画像やテキストなど、**Screen Library** からのアイテムをレイアウトする際に **Define Screen** ページ内でこのグリッドを使用します。
Visible（可視）	クリックしてこのグリッドが表示されるようにします。
Snap to grid（グリッドにスナップ）	クリックしてアイテムの **Reference point**（基準点）をグリッドにスナップします。
File transparency（ファイルの透明性）	クリックして、スタックしたスクリーンエリアに渡って全アイテムが見えるようにします。 異なるスクリーンエリアにあるアイテムを揃えるときにこのオプションを使用できます。
Color（色）	クリックして、ドロップダウンリストからグリッドのラインの色を選択します。
Cell width（セル幅）	クリックして、グリッドライン間の水平距離（画素数）を設定します。
Cell height（セルの高さ）	クリックして、グリッドライン間の垂直距離（画素数）を設定します。
Layout pane（レイアウトペイン）	各スクリーンエリアに表示されるレイアウトペインのグリッドの特徴を設定します。 スクリーンエリアのレイアウト時に **Define Areas** ページ内でこのグリッドを使用します。
Visible（可視）	クリックしてグリッドの線を見えるようにします。
Snap to grid（グリッドにスナップ）	スクリーンエリアの左上角でグリッドにスナップします。
Color（色）	クリックして、ドロップダウンリストからのグリッドのラインの色を選択します。

Options ウィンドウ　Grid Settings

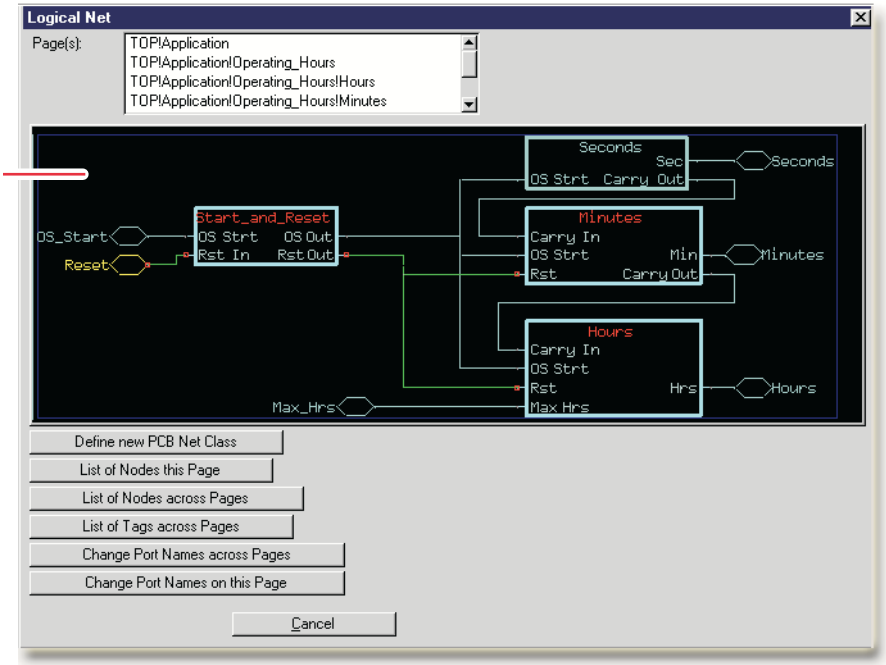
Item	Description
Cell width（セル幅）	クリックして、グリッドライン間の水平距離（画素）を設定します。
Cell height（セルの高さ）	クリックして、グリッドライン間の垂直距離（画素）を設定します。

レイアウトペインとスクリーンエリアグリッド



Logical Net ウィンドウ

Page Preview pane—highlights a selected route and its connections

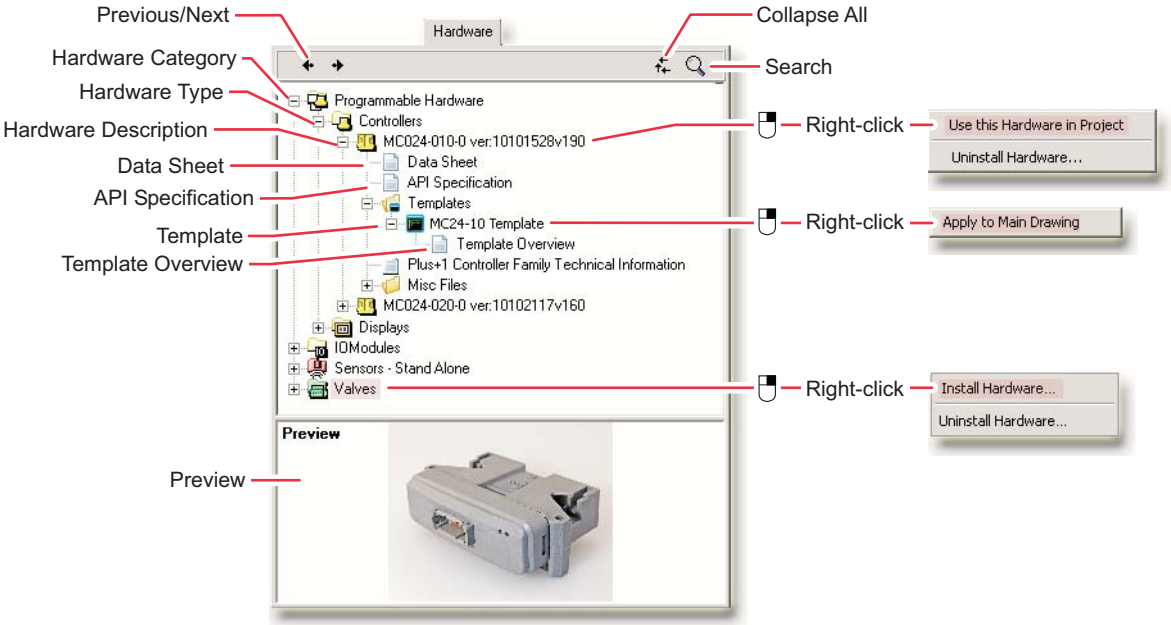


このウィンドウを用いて単一のページ内あるいは複数のページにまたがる配線の接続をトレースすることができます。

Logical Net ウィンドウ

項目	説明
Page(s)	選択したルートから接続されているページのリストを表示します。 **Page(s)**にリストされているページ名をクリックして Page Preview ペインにそのページを表示します。
Page Preview pane	**Page(s)**リストで選択したページをプレビューし、選択した配線と接続をハイライトします。 このペイン内をクリックすると **View Logical Net** ウィンドウが閉じ、プレビューしていたページが図面エリアに表示されます。
List of Nodes this Page	このリリース版では未実装です。
List of Nodes across Pages	このリリース版では未実装です。
List of Tags across Pages	このリリース版では未実装です。
Change Port Names across Pages	クリックして **Define New Name for <*n*> Ports** ウィンドウを表示します（*<n>*は名前を変更するポートの数）。 選択した配線に属するポート名は全てこのウィンドウで入力した名前で置き換えられます。 **Define New Name for <*n*> Ports** ウィンドウは選択した配線が現れる各々のページのポート名を置き換えます。 このウィンドウではページに信号を引き込むポートやページから信号を引き出すポートの名称は変更しません。
Change Port Names on this Page	クリックして **Define new Port Name** ウィンドウを表示します。 このウィンドウで入力した名前は、Page Preview ペインでハイライトされた配線内のポート名のみを置き換えます。 **Define new Port Name** ウィンドウは Page Preview ペインで表示されるページ内のポート名のみを置き換えます。 このウィンドウではページに信号を引き込むポートやページから信号を引き出すポートの名称は変更しません。

Hardware タブ



このタブは、ハードウェアカテゴリ、ハードウェアのタイプ、ハードウェアのモデル、ハードウェア記述によって分類されたハードウェア関連項目を含みます。PLUS+1 ハードウェアモデルごとにハードウェア記述が存在します。

- ハードウェア記述には、選択したハードウェアモデルのアプリケーションをコンパイルするのに PLUS+1 GUIDE プログラムで必要なリソースを含んでいます。ハードウェア記述を **Hardware** タブから **Project Manager** タブにドラッグすることによって、このリソースが PLUS+1 GUIDE プログラムで使えるようになります。
- 各ハードウェアの関連事項には、ハードウェア **Template** およびハードウェアに関するドキュメンテーションがあります。

プログラミングを簡単にするために、ハードウェア **Template** には、選択ハードウェアに適合する予め定義された入力、出力、メモリ位置が含まれます。ハードウェア **Template** を **Hardware** タブから図面エリアにドラッグすることによってハードウェア **Template** が使えるようになります。

ハードウェアドキュメンテーションには通常、仕様、アプリケーションインターフェイス(API)情報、ハードウェアモデルのテンプレートが含まれます。

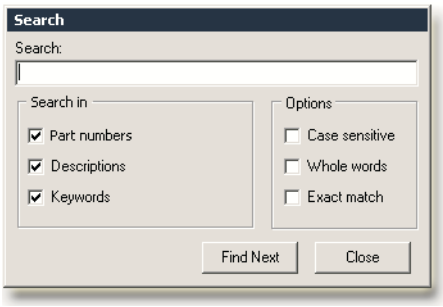
Hardware タブ

項目	説明
Previous/Next（前／次）	クリックして **Hardware** ツリーのブランチを展開します。
Collapse All（全て隠す）	クリックしてハードウェアツリーのブランチを全て隠します。
Search（検索）	クリックして **Search** ウインドウを表示します。 このウインドウを使用して、**Hardware**ツリー内の項目を検索します。詳しくは、*71*ページの*Search* をご覧ください。
Hardware Category （ハードウェアカテゴリ）	これらのフォルダでは、ハードウェアを **Programmable Hardware** や **IO Modules** のようなカテゴリ毎に分類しています。

Hardware タブ

項目	説明
Hardware Type （ハードウェアのタイプ）	これらのフォルダでは、ハードウェアをハードウェアカテゴリ内でタイプ別に分類しています。
Hardware Description （ハードウェア記述）	特定の PLUS+1 ハードウェアモデル用のアプリケーションを作成・コンパイルするために PLUS+1 GUIDE プログラムが必要とするリソースにリンクします。 これらのリソースをハードウェア記述を **Hardware** タブから **Project Manager** タブにドラッグすることによってプログラムで使用できるようにします。 ハードウェア記述を右クリックし、**Use this hardware in project** や **Uninstall Hardware** のコマンドがあるポップアップメニューを表示します。
Data Sheet（データシート）	クリックして、ハードウェアモデルの製品仕様書を表示します。この文書は、製品の特長、寸法、電力供給、ピンの割り当てなどハードウェアモデルの一般的な概要を示します。
API Specification（API 仕様）	クリックしてハードウェアモデルのアプリケーションインターフェイス（API)記述書を表示します。 ハードウェアの入力装置と出力装置を構成するときにこの文書を参照します。
Template（テンプレート）	クリックしてハードウェアモデルのプログラミングテンプレートを表示します。 プログラミングを簡単にするために、**Template** は選択されたハードウェアモデルに適合する、予め定義された入力、出力、メモリ位置を含みます。 ハードウェア **Template** を **Hardware** タブから図面エリアにドラッグすることによってハードウェア **Template** が使えるようになります。
Template Overview （テンプレート概要）	クリックしてテンプレートの使用方法についての基本情報を示す文書を表示します。
Preview（プレビュー）	選択したハードウェアモデルのプレビューを示します。 **Option**ウインドウの設定（***58***ページ参照）を使用して、このプレビュー機能を有効にします。

Search ウィンドウ

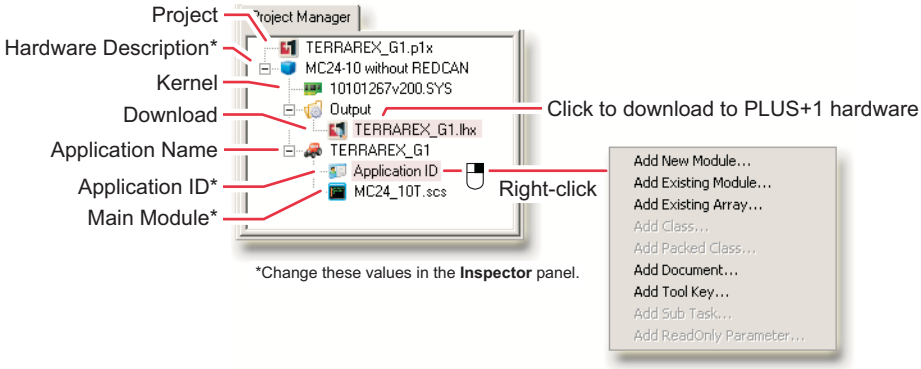


このウインドウを使用して、特定のハードウェア記述の **Hardware** タブを検索します。

Search ウインドウ

項目	説明
Search（検索）	検索用語をここに入力します。
Search in（検索詳細）	**Part numbers**（部品番号）： **Search** に入力した番号に適合する部品番号を検索するときにチェックを入れます。 Descriptions（説明）： **Search** に入力した用語に適合する説明を検索するときにチェックを入れます。 Keywords（キーワード）： **Search** に入力した用語に適合するキーワードを検索するときにチェックを入れます。
Options（オプション）	この **Options** 項目を使用して、検索条件を絞り込みます。 **Case sensitive**（大文字／小文字を区別する）： チェックを入れると、**Search** に入力した用語の大文字／小文字が一致した項目のみを検索します。 **Whole words**（すべての用語を含む）： チェックを入れると、**Search** に入力した用語の全てを含む項目のみを検索します。 **Exact match**（完全一致）： チェックを入れると、**Search** に入力した用語に完全に一致する項目のみを検索します。

Project Manager タブ



このタブは、インストールしたハードウェア記述のリストを表示します。これらのハードウェア記述は個々の PLUS+1 ハードウェアモデルに固有のものです。ハードウェア記述には特定の PLUS+1 ハードウェアモデル用のアプリケーションを作成・コンパイルするために PLUS+1 GUIDE プログラムが必要とするリソースが含まれています。

このハードウェア記述を PLUS+1 GUIDE プログラムで使用できるようにするためには、ハードウェア記述を **Hardware** タブから **Project Manager** タブにドラッグします。

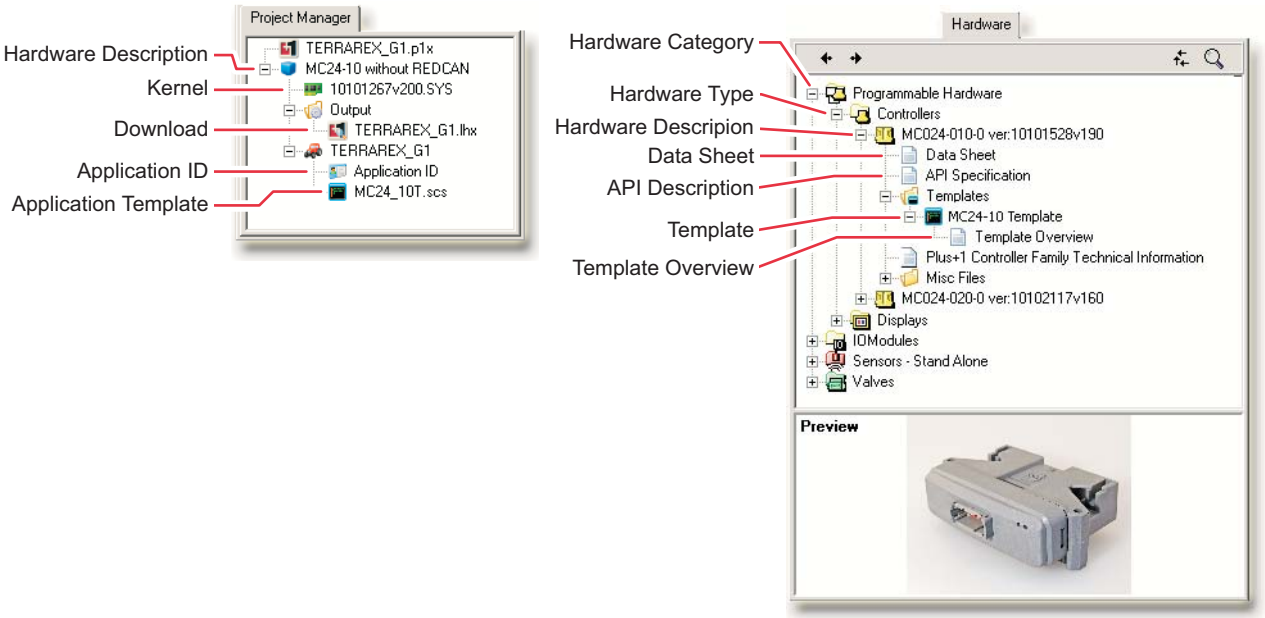
Project Manager タブ

項目	説明
Project	プロジェクト名
Hardware Description （ハードウェア記述）	ハードウェア記述を **Hardware** タブからドラッグしたときにインストールしたハードウェア記述を表示します。
Kernel（カーネル）	このハードウェアで使用するオペレーティングシステム。このカーネルはハードウェア記述の一部であり、変更できません。
Download（ダウンロード）	ダウンロードファイルの名前。 アプリケーションをコンパイル後、このファイルをクリックします。PLUS+1 サービスツールプログラムが開き、このファイルのダウンロード準備が完了します。
Application Name **（アプリケーション名）**	アプリケーション名。 1 つのアプリケーションはメインモジュールと複数の補助モジュールから構成することができます。 アプリケーション名を右クリックして以下のコマンドのポップアップメニューを表示します。 **Add New Module**（新しいモジュールを追加）： クリックして新しいモジュールをアプリケーションに追加します。 **Add Existing Module**（既存のモジュールを追加）： クリックして **Select File** ウインドウを表示します。このウインドウを使用して、既存のモジュールを選択し、アプリケーションに追加します。 **Add Existing Array**（既存のアレイを追加）： クリックして **Select File** ウインドウを表示します。このウインドウを使用して、既存のアレイをアプリケーションに追加します。詳細については、194 ページのファイルからのアレイ定数をご覧ください。 **Add Document**（文書の追加）： クリックして **Select File** ウインドウを表示します。このウインドウを使用して、プロジェクトフォルダに追加したいワードや PDF などのファイルを選択します。
Application ID **（アプリケーション ID）**	アプリケーション記述を置き換えます。この記述はオプションです。**Inspector** タブを使用してアプリケーション記述に入ります。 アプリケーションをアップデートする場合、ダウンロード開始可能にするには古い記述と新しい記述を一致させる必要があります。

Project Manager タブ

項目	説明
Main Module	クリックして図面エリアのモジュールを表示します。 一般的にこのモジュールは **Template** を **Hardware** タブからドラッグすることによって置き換え可能です。 **Template** には、選択した PLUS+1 ハードウェアに適合する、予め定義した入力、出力、メモリ位置があります。 アプリケーション名を右クリックすると、従属モジュール追加のコマンドがあるポップアップメニューが表示されます。

Project ManagerタブとHardwareタブについて



右側の **Hardware** タブには、PLUS+1 GUIDE プログラムが PLUS+1 ハードウェアを作成・コンパイルするのに必要なハードウェア関連リソースがあります。

- **Hardware** タブは一般的に主に以下の項目を含んでいます。
  - PLUS+1 GUIDE プログラムが特定 PLUS+1 ハードウェアモデルのアプリケーションを作成・コンパイルするのに必要なリソースを含むハードウェア記述。

  一般に、各ハードウェア記述と関連している以下のもの。
  - ハードウェアの説明をしているハードウェアデータシート
  - ハードウェアでの入力と出力の構成方法を示す API（アプリケーションインターフェイス）仕様
  - アプリケーションを作成したハードウェアに適合する、定義済みの入力、出力、メモリ位置を持つ **Template**
  - **Template** の使用方法を述べている **Template Overview**
- **Project Manager** タブの主な項目は以下の通りです。
  - PLUS+1 GUIDE プログラムが特定の PLUS+1 ハードウェアモデルのアプリケーションを作成・コンパイルするのに必要なリソースを含む **Hardware Description**（ハードウェア記述）。**Hardware** タブから **Project Manager** タブへハードウェア記述をドラッグします。
  - プログラミング中のハードウェアが必要とするカーネル（オペレーティングシステム）。カーネルはハードウェア記述に沿ってインストールを行います。

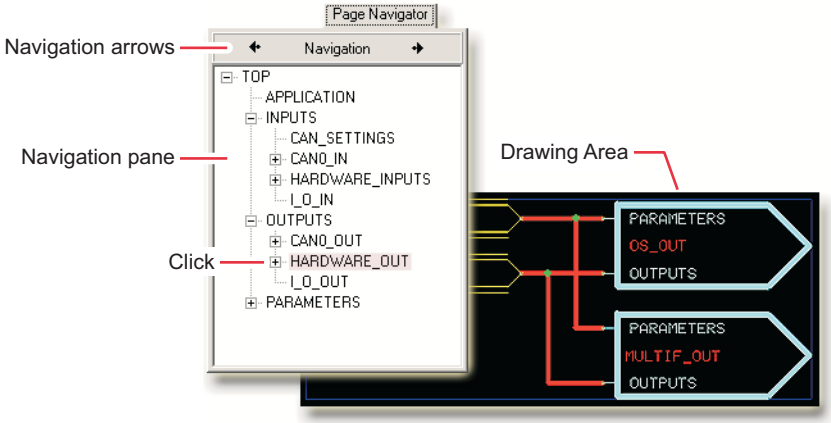
- Limited hex (*.lhx)フォーマットのダウンロードファイル。PLUS+1 GUIDE プログラムで作成アプリケーションをコンパイルした後、このファイルをハードウェアにダウンロードします。

  ダウンロードを特定のハードウェアに限定でき、またアプリケーションコードを隠すこともできます。

- アプリケーションを記述するアプリケーション ID（オプション）。アプリケーションのアップデート時、ダウンロード開始前に新しい ID と古い ID を一致させてください。

- アプリケーションテンプレート。ここからアプリケーションの作成を開始します。**Template** を **Hardware** タブからドラッグして、図面エリアにドロップすることによって、このアプリケーションテンプレートをインストールします。

Page Navigator Tab

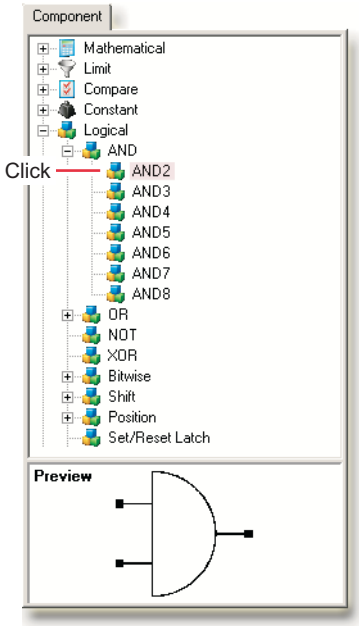


このタブを使用して、直接アプリケーションのページに進みます。

Page Navigator タブ

項目	説明
Navigation arrows（矢印）	クリックして、図面エリアで前回のナビゲーション選択を表示します。
Navigation pane（ナビゲーションペイン）	アプリケーション内の全ページのツリービューを表示します。 ツリー内のページ名をクリックして、図面エリアにページを表示します。

Component タブ



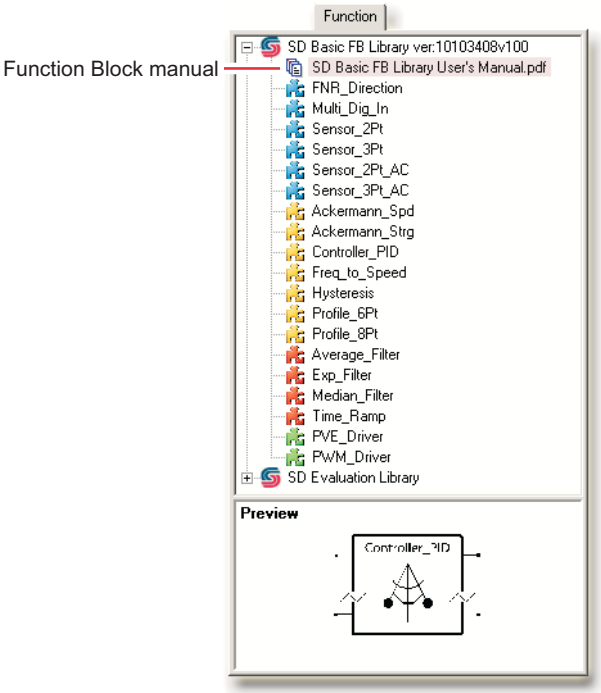
このタブは、アプリケーションで使用するために図面エリアにドラッグできるコンポーネントのツリービューを表示します。

コンポーネントは、アプリケーションの基本構築ブロックです。

Component タブ

項目	説明
Components **（コンポーネント）**	アプリケーションで使用できるコンポーネントをタイプ別にリストアップします。 必要なコンポーネントをここから図面エリアにドラッグします。 新しいコンポーネントライブラリを入手したときは、**Setup** メニューの **Install Library**（ライブラリのインストール）コマンドを使用して、そのコンポーネントライブラリをこのタブに追加します。
Preview（プレビュー）	クリックしたコンポーネントのプレビューを示します。 **Option**ウインドウの設定（***58***ページ参照）を使用して、プレビューを表示したり、隠したりします。

Function タブ



このタブは、アプリケーションで使用するために図面エリアにドラッグできるファンクションのツリービューを表示します。

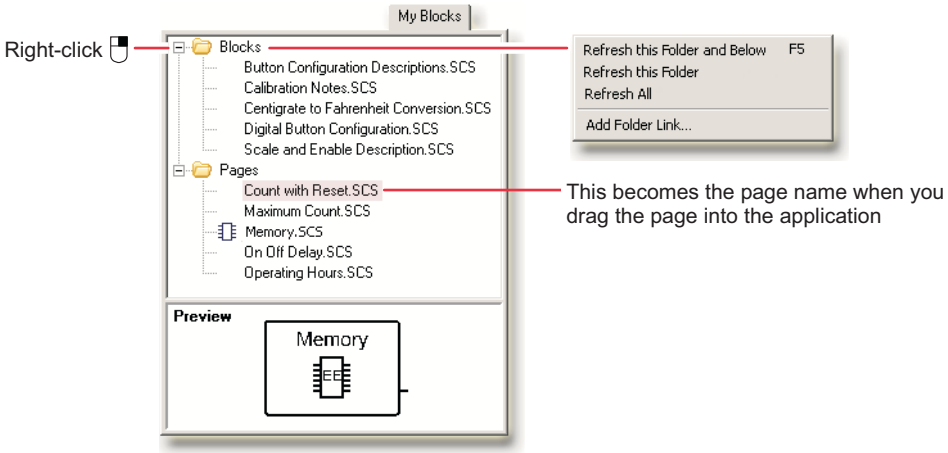
ファンクションとは、ザウアーダンフォス社が作成したコンポーネントのブロックで、例えば以下のような特定のタスクができるように設計されたものがあります。

- アッカーマンステアリング
- 周波数–速度変換

Function タブ

項目	説明
Function Block manual	クリックして基本ファンクションブロックライブラリのマニュアルを表示します。
Functions	アプリケーションで使用できる機能をタイプ別にリストしています。 必要な機能をここから図面エリアにドラッグします。 新しいファンクションライブラリが使用できるときは、**Setup** メニューの **Install Library** コマンドを使用して、ファンクションライブラリをこのタブに追加します。
Preview	選択した機能のプレビューを示します。 **Option**ウインドウの設定（***58***ページ参照）を使用して、プレビューを表示したり、隠したりします。

My Blocks タブ



このタブを使用すると、作成したブロックやページをプロジェクト内で頻繁に使用する場合に、ブロックやページの選択が迅速に行えます。選択したブロックやページは全てのプロジェクトで使用できます。

- **Blocks** フォルダと **Pages** フォルダは、このタブにブロックとフォルダを保存します。これらのフォルダへのパスは、通常、C:\Documents and Settings\<User Name>\My Documents\Sauer-Danfoss\PLUS1\GUIDE\MyBlocks です。
- **Add Folder Link** コマンドを使用してブロックやページ用のカスタムフォルダを作成します。

My Blocks タブ

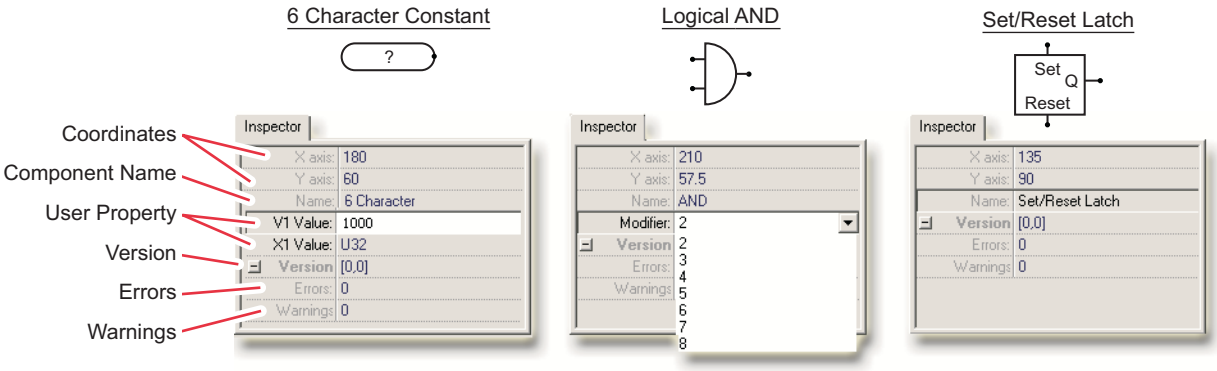
項目	説明
Blocks	**File** メニューの **Block > Export Block** コマンドを使用して **Blocks** フォルダへエクスポートしたブロックのリストを表示します。 図面エリアで使用したいブロックをクリックしてドラッグします。
Pages	**File** メニューの **Page > Export Page** コマンドを使用して **Pages** フォルダへエクスポートしたページのリストを表示します。 図面エリアで使用したいページをクリックしてドラッグします。
Preview	クリックしたブロックやページのプレビューを表示することができます。

Preview ペイン画像やアイコンを作成するには、以下の手順を行ってください。

1. ブロックやページの Windows Metafile (*.WMF)画像を作成します。この画像は、**Preview** ペインに表示されます。画像サイズは、150 x 150 ピクセルより小さくしてください。
2. ブロックやページに使用したのと同じ名前を使って、**Blocks** フォルダや **Pages** フォルダに WMF ファイルを保存してください。
3. ブロックやページのアイコン(*.ICO)を作成します。このアイコンは、**My Blocks** タブのツリー内のブロック名やページ名の隣に表示されます。アイコンサイズは、16 x 16 ピクセルより小さくしてください。
4. ブロックやページに使用したのと同じ名前を使って、**Blocks** フォルダや **Pages** フォルダに ICO ファイルを保存してください。

たとえば、***Memory.WMF*** ファイルおよび ***Memory.ICO*** ファイルは、***Memory.SCS*** ページのプレビューとアイコンを示します。

Inspector タブ



このタブを使用して、**Query/Change** ツールで選択したアイテムのプロパティを確認し、変更します。

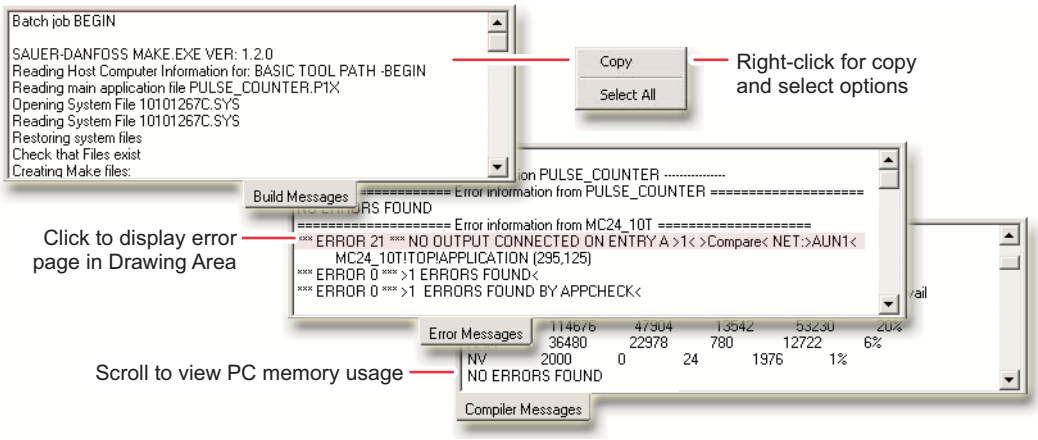
値を変更するのに、小さいポップアップ **Edit Value** ウインドウを使用することもできます（425 ページ参照）。

選択したアイテムの型によって、変更できるプロパティが決まります。

Inspector タブ

項目	説明
Coordinates（座標）	アイテムを置いたページのx軸座標とy軸座標。
Name（名前）	アイテムの名前。
User Property（ユーザープロパティ）	プロパティを変更するために入力した値です。 変更できるプロパティは、選択したアイテムに依存します。 上の図は、プロパティの 3 つの例を示しています。 **6-Character** 定数コンポーネント用 **Inspector** タブは、コンポーネント値とそのデータ型を設定する **V1** と **X1** の値を表示します。 **AND** コンポーネント用の **Inspector** タブはこのコンポーネント上の入力数を設定する **Modifier** 値を示します。 **Set/Reset Latch**（セット／リセット　ラッチ）コンポーネント用 **Inspector** タブは、変更できる値がありません。
Version（バージョン）	アイテムのリリースバージョン。
Errors（エラー）	アイテムのエラー状態を示します。 **0** = エラーなし。PLUS+1 GUIDE プログラムは、正常にアイテムをコンパイルします。 **1** = エラーあり。PLUS+1 GUIDE プログラムはアイテムをコンパイルできません。コンパイルプロセスがエラーにより中止されます。アイテムを新しくする必要があります。
Warnings（警告）	アイテムの警告状態を示します。 **0** = 警告なし。PLUS+1 GUIDE プログラムは、警告を受けずに、アイテムをコンパイルします。 **1** = 警告あり。PLUS+1 GUIDE プログラムはコンパイルしますが、警告付きです。できるだけ早く、アイテムを新しくする必要があります。

Compiler Messages タブ



Compiler Messages ペインには、**Build Messages** タブ、**Error Messages** タブ、**Compiler Messages** タブがあります。

これらのタブは次のような目的に使用します。

- ダウンロードファイルを作成するコンパイルプロセスを監視する。
- コンパイルプロセスを中止するエラーを見つけて、確定する。

コンパイルプロセスが中断したときは、まず **Error Messages** タブでエラーをチェックします。次に **Compiler Messages** タブと **Build Messages** タブでエラーをチェックします。

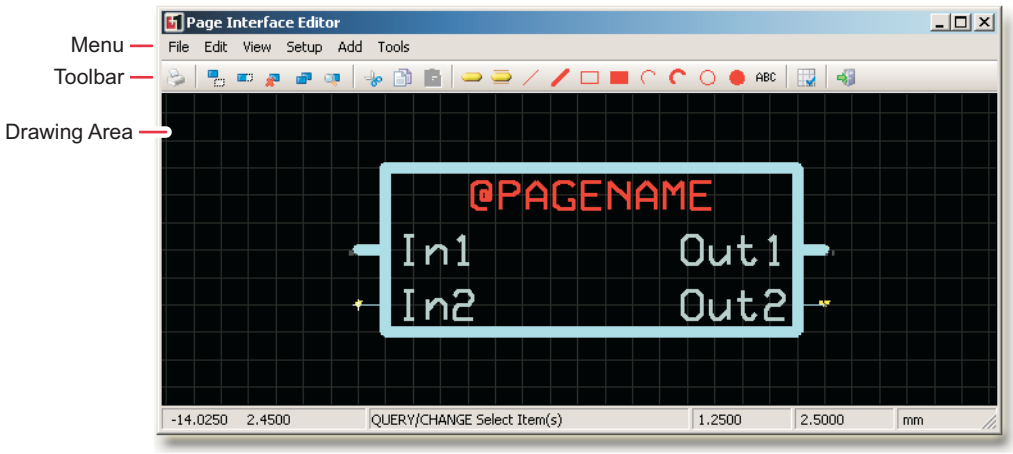
Compiler Messages タブ

項目	説明
Build Messages タブ	コンパイラは、アプリケーションをダウンロード可能ファイルにコンパイルしながら、このタブにビルドメッセージを表示します。
Error Messages タブ	コンパイラは、このタブにコンパイルのエラーメッセージを表示します。エラーはコンパイルプロセスを中止します。 アプリケーション内の全てのエラーと多くのプロセスエラーにより、このタブにメッセージが表示されます。コンパイルプロセスが中止した場合はまずタブを確認してください。 アプリケーション内のエラーについてのメッセージに、エラーの種類とコンパイラがエラー確認時のページと座標を示す行があります。この行をクリックして、図面エリア内のエラーソースページを表示します。 （コンパイラがエラーを見つけたページとエラー原因のページが必ずしも同じとは限りません。エラーソースを確定するには、上位のページに戻る必要が出てくる場合もあります。） Project フォルダ内の ***.err** ファイル（* = プロジェクト名）にエラーメッセージのログが記録されています。
Compiler Messages タブ	コンパイラは、プロジェクトフォルダ内にある ***Screen.tmp*** ファイルに記録したメッセージを表示します。

Page Interface Editor ウィンドウ

ツールバー >

または **Add** メニュー > **Page Interface Editor**



Page Interface Editor を用いてページのトップビューを編集します。

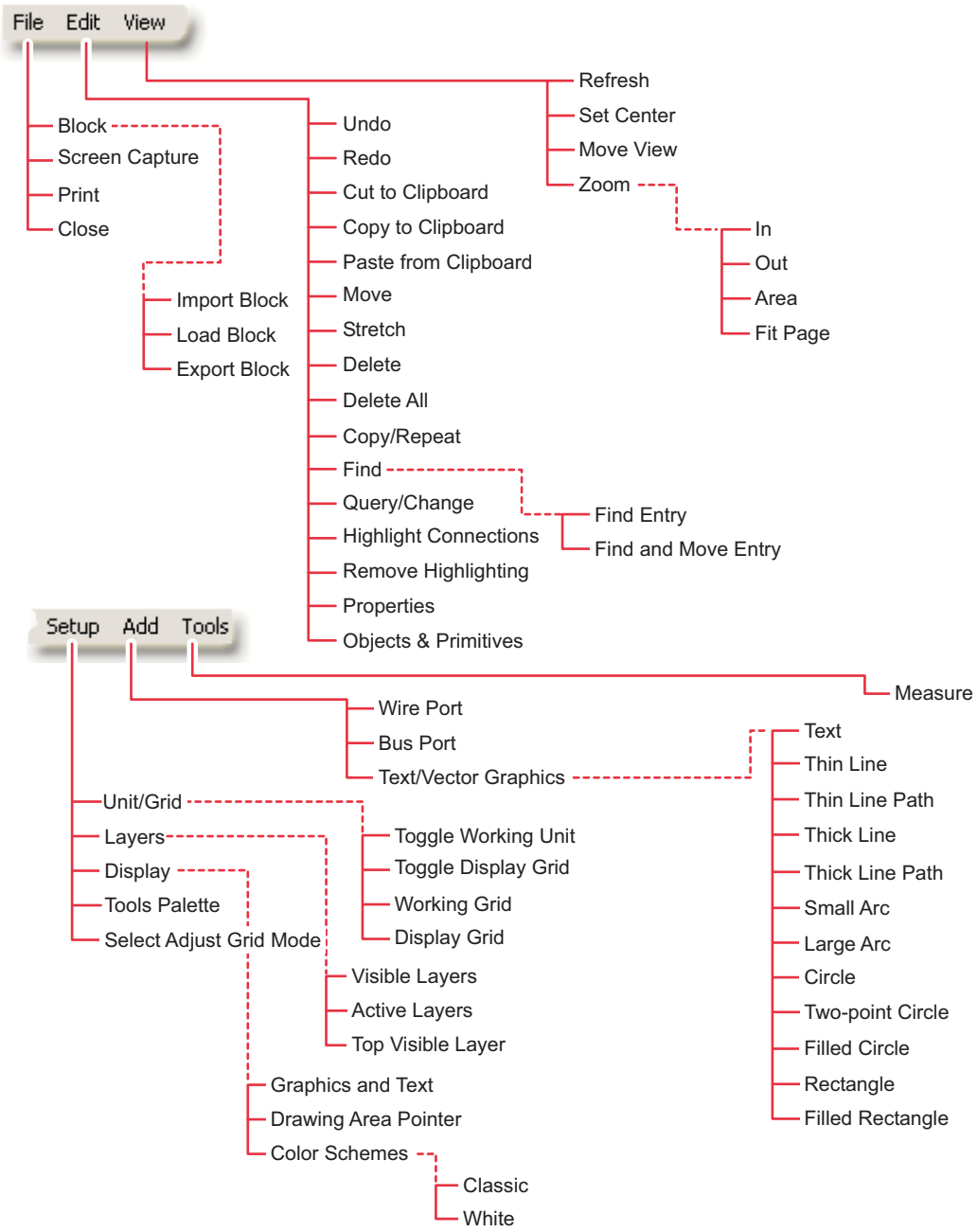
@PAGENAME プレースホルダの **Layer** 値は **PageName** のままにしておいてください。この値を変更するとページ名が消失することがあります。また、コンパイルエラーや PLUS+1 プログラムの応用機能に支障をきたす場合があります。

ページ、ページのトップビュー、Page Interface Editorの表示に関しては*91*ページの*ページ、ページのトップビュー、ページインタフェースエディタのウィンドウについて*を参照してください。

Page Interface Editor ウィンドウ要素

項目	説明
Menus	メニューバーを用いて **Page Interface Editor** ウィンドウコマンドにアクセスします。
Toolbar	ツールバーを用いてよく使われる **Page Interface Editor** ウィンドウコマンドにアクセスします。
Drawing Area	ページのトップビューをここで編集します。

Page Interface Editor ウィンドウメニュー



メニューバーを用いて **Page Interface Editor** ウィンドウコマンドにアクセスします。

Page Interface Editor ウィンドウツールバーにあるボタンにはよく使うコマンドを載せています。これらのボタンに関しては*88*ページの *Page Interface Editor ウィンドウツールバー* を参照してください。

Page Interface Editor ウィンドウメニュー

項目	説明
File	このメニュー内のコマンドで **Page Interface Editor** ウィンドウファイルを管理します。
Block	**Import Block—Select Import Preview** ウィンドウを表示します。 このウィンドウを使用して Page Interface Editor ウィンドウの図面エリアにインポートする **DXF ASCII** フォーマットのファイルを選択します。 **Load Block—Schematics Symbol Editor** ウィンドウを表示します。 このウィンドウを使用して SCS フォーマットファイルを選択します。ファイル内のエントリーやグラフィックなどのトップビューアイテムを **Page Interface Editor** ウィンドウの図面エリアにインポートします。 （Export Block コマンドは Page Interface Editor ウィンドウの図面エリアで選択されたアイテムをエクスポートします。） **Export Block— Page Interface Editor** ウィンドウの図面エリアからエクスポートするアイテムを選択した後、**Symbol Block Binary Export** ウィンドウを表示します。 **Symbol Block Binary Export** ウィンドウを使用して選択したアイテムを SCS ファイルにエクスポートします。
Screen Capture	**Page Interface Editor** ウィンドウの図面エリア内のアイテムを選択した後、**Select Screen Dump Format** ウィンドウを表示します。 このウィンドウを使用して、選択したアイテムを印刷するか、BMP や TIFF ファイルで保存するか、BMP や Metafile フォーマットでクリップボードにコピーするかを選びます。
Print	**Print** ウィンドウを表示します。 このウィンドウを使用して **Page Interface Editor** ウィンドウの図面エリアの内容を印刷します。
Close	**Page Interface Editor** ウィンドウを閉じます。
Edit	このメニュー内のコマンドで **Page Interface Editor** ウィンドウの図面エリア内のアイテムに変更を加えます。
Undo	プログラミング操作を元に戻します。 コンピュータのメモリに応じて最大 10 操作まで元に戻すことができます。
Redo	**Undo** 操作を元に戻します。 コンピュータのメモリに応じて、最大 10 操作まで **Undo** コマンドを元に戻すことができます。 **Options** ウィンドウによって Undo/Redo 機能を有効化/無効化することができます。このウィンドウを表示するには **Setup** メニューで **Options** をクリックします。
Cut to Clipboard	**Page Interface Editor** ウィンドウの図面エリア内で選択したアイテムを削除し、それらをクリップボードにコピーします。
Copy to Clipboard	**Page Interface Editor** ウィンドウの図面エリア内で選択したアイテムをクリップボードにコピーします。
Paste from Clipboard	クリップボードの内容を **Page Interface Editor** ウィンドウの図面エリアにペーストします。
Move	**Page Interface Editor** ウィンドウの図面エリア内で選択したアイテムを移動します。
Stretch	選択した頂点を移動することで線や円弧などのグラフィックアイテムをストレッチ（伸縮）します。 選択した文字列やエントリーアイテムを移動します。
Delete	**Page Interface Editor** ウィンドウの図面エリア内で、選択したアイテムを削除します。選択したアイテムは白くなります。 独立したアイテムや同一アイテムを選択すると **Attributes**（属性）ウィンドウが表示されます。 **Attributes** ウィンドウで **OK** をクリックして選択アイテムを削除します。 **Select Item Class** ウィンドウは、複数の異なるアイテムを選択するときに表示されます。 このウィンドウを使用して、削除したいアイテムを選択します。アスタリスク (*) で削除するアイテムを指定、ダッシュ (-) で削除しないアイテムを指定します。 **Select Item Class** ウィンドウで **OK** をクリックして選択アイテムを削除します。
Delete All	**Delete all items?** メッセージを含む **Question** ウィンドウを表示します。 **Yes** をクリックして **Page Interface Editor** ウィンドウの図面エリア内の全てのアイテムを削除します。

Page Interface Editor ウィンドウメニュー

項目	説明
Copy/Repeat	**Page Interface Editor** ウィンドウの図面エリア内で、選択したアイテムをコピーします。選択したアイテムは白くなります。 クリックしてコピーしたアイテムを置きます。Esc を押すとコピーを中止します。 このコマンドは **Page Interface Editor** ウィンドウの図面エリア内でのみ有効です。
Find	**Find Entry: Page Interface Editor** ウィンドウの図面エリア内の各 **Entry**（ポート）をリストする **Select Entry** ウィンドウを表示します。 **Select Entry** ウィンドウ内のポート名をクリックすると、選択したポートを **Page Interface Editor** ウィンドウの図面エリアでズームインして中央に表示します。 **Find and Move Entry: Page Interface Editor** ウィンドウの図面エリア内の各 **Entry**（ポート）をリストする **Select Entry** ウィンドウを表示します。 **Select Entry** ウィンドウ内でポート名をクリックすると、選択したポートを **Page Interface Editor** ウィンドウの図面エリア内のクリックした場所に移動します。
Query/Change	**Page Interface Editor** ウィンドウの図面エリア内で、アイテムのプロパティ（テキスト、テキストのサイズ、色、線幅など）を変更するときに使用します。 プロパティを変更したいアイテムをクリックしてください。選択したアイテムに適したダイアログボックスが表示されます。このウィンドウ内でプロパティを変更します。
Highlight Connections	**Page Interface Editor** ウィンドウの図面エリア内で、選択したアイテムをハイライトします。
Remove Highlighting	**Page Interface Editor** ウィンドウの図面エリア内で、ハイライトをキャンセルします。
Properties	上級ユーザのみ使用できます。
Objects & Primitives	**Select Item Class** ウィンドウに続いて **Detail Data Base** ウィンドウを表示します。 **Detail Data Base** ウィンドウは **Page Interface Editor** ウィンドウの図面エリア内の全ての編集可能なアイテムのリストを表示します。アイテムは通常 **Entry** や **Text** などのタイプ、および **PortName** や **SymbolBodyText** などのレイヤーによって区別されます。 **Select Item Class** ウィンドウを使用して **Detail Data Base** ウィンドウ内に表示されるアイテムのリストを絞り込むことができます。

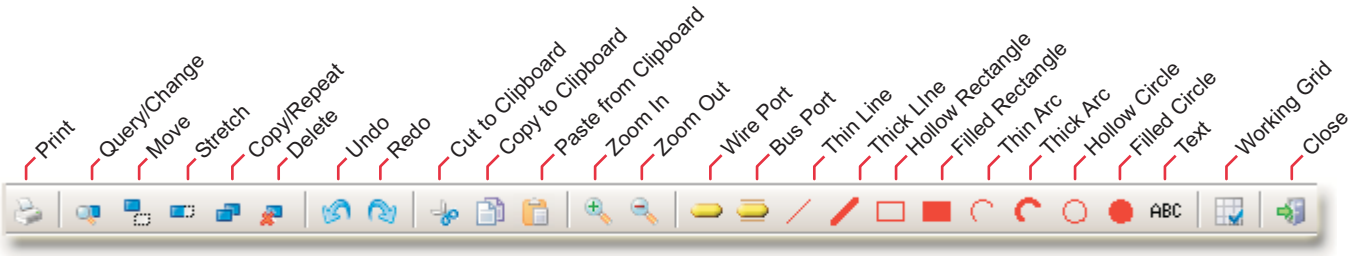
Page Interface Editor ウィンドウメニュー

項目	説明
View	このメニュー内のコマンドは **Page Interface Editor** ウィンドウの図面エリア内のビューを変更します。
Refresh	**Page Interface Editor** ウィンドウの図面エリアを再描画します。
Set Center	ポインタでクリックした場所を **Page Interface Editor** ウィンドウの図面エリアの中央に移動します。
Move View	白い境界線を持つ移動可能で透明な長方形を表示します。クリックすると **Page Interface Editor** ウィンドウの図面エリアがこの長方形の境界に入るように中心が移動します。
Zoom	**In**: ズームインして、ポインタをクリックした場所を **Page Interface Editor** ウィンドウの図面エリアの中央に移動します。 **Out**: ズームアウトして、ポインタをクリックした場所を **Page Interface Editor** ウィンドウの図面エリアの中央に移動します。 **Area**:**Page Interface Editor** ウィンドウの図面エリアを 2 箇所クリックしたポイントで決まる範囲にズームします。 **Fit Page**: ページ内の全てのアイテムが **Page Interface Editor** ウィンドウの図面エリア内にフィットするようにビューのサイズを変更します。
Setup	このメニュー内のコマンドは **Page Interface Editor** ウィンドウの作業環境を設定します。
Unit/Grid	**Toggle Working Unit:** クリックして測定単位をインチとミリメートルの間で切替えます。 **Toggle Display Grid:** グリッド線の表示・非表示を切替えます。 表示グリッド(Display Grid)とは **PLUS+1 GUIDE**、**Page Interface Editor**、**Module Viewer** の図面エリアで表示されるグリッドのことです。 **Working Grid: Define Work Grid** ウィンドウを表示します。 このウィンドウを使用して作業グリッドのサイズ(グリッド線の間隔)を定義します。 **PLUS+1 GUIDE**、**Page Interface Editor**、**Module Viewer** ウィンドウの図面エリア内のアイテムはこの作業グリッドにスナップします。これは目に見えないグリッドです。 **Display Grid: Define Display Grid** ウィンドウを表示します。 このウインドウを使用して表示グリッドの間隔を定義します。 G を押して新しく選択した表示グリッドを表示します。
Layers	**Visible Layers: Define Visible/Invisible Objects/Layers** ウィンドウを表示します。 このウィンドウを使用して **Page Interface Editor** ウィンドウの図面エリア内でレイヤー、プリミティブ(図面オブジェクトと文字列)、エントリーの表示をコントロールします。 **Active Layers: Define Active/Inactive Objects/Layers** ウィンドウを表示します。 このウィンドウを使用して **Page Interface Editor** ウィンドウの図面エリア内でアクティブなレイヤー、プリミティブ(図面オブジェクトと文字列)、エントリーを定義します。 アクティブなアイテムは編集可能、アクティブでないアイテムは編集不可能です。 **Top Visible Layer: Top Visible Layer** ウィンドウを表示します。
Display	**Graphics and Text**: **Define Drawing Mode Filled Area & Text** ウインドウを表示します。 このウインドウを使用して **Page Interface Editor** ウィンドウの図面エリア内のアイテムの作図および表示モードを変更します。 デフォルト設定値は全アイテムについて **True Size** です。 **Drawing Area Pointer**: **Graphics Pointer Setup** ウインドウを表示します。 このウインドウを使用して、**Page Interface Editor** ウィンドウの図面エリアについてポインタ表示、ポインタ解像度の変更および、x軸と y 軸の0基準の設定をします。 **Color Schemes**: **Classic** スキームを選択すると **Page Interface Editor** ウィンドウの図面エリアの背景を黒にします。**White** スキームを選択すると **Page Interface Editor** ウィンドウの図面エリアの背景をオフホワイトに設定します。
Tools Palette	Displays the **Icon Menu** window.

Page Interface Editor ウィンドウメニュー

項目	説明
Select Adjust Grid Mode	**Setup Select Items Grid Adjust Mode** ウィンドウを表示します。 このウィンドウを使用して、新たに作業グリッドサイズ（グリッド線の間隔）を異なる値に変更した後の **Page Interface Editor** ウィンドウの図面エリア内のアイテムのスナップ移動の振る舞いを設定します。 **Always adjust reference to working grid:** アイテムを古いグリッドの位置から新しく設定したグリッドサイズのステップで移動します。下の*例*を参照してください。—Moves an item from its position in the old grid in steps set by the size of the new grid. See the ***Example*** below. **Ask to adjust single reference to working grid:** アイテムを移動するときに **Question** ウィンドウで次の選択肢を表示します: **Force XY to grid? Yes/No** —Moving an item displays a **Question** window with the following text: **Force XY to grid? Yes/No**. **Yes** を選択するとアイテムを新しい作業グリッドの交点に移動します—Moves the item to an intersection in the new working grid. **No** を選択するとアイテムを古いグリッドの位置から新しく設定したグリッドサイズのステップで移動します。下の*例*を参照してください。—Moves the item from its position in the old grid in steps set by the size of the new grid. See the ***Example*** below. **Adjust only multiple reference to working grid:** 単独のアイテムを新しい作業グリッドの交点に移動します。複数のアイテムは古いグリッドの位置から新しく設定したグリッドサイズのステップで移動します。—Moves a single item to an intersection in the new working grid. Moves multiple items from their positions in the old grid in steps set by the size of the new grid. **Adjust only multiple reference to working grid** がデフォルトの設定です。 *例*—古いグリッドのサイズが 0.25 mm; 新しいグリッドのサイズが 1.0 mm のとき、0.25 mm 上のアイテムを 1.0 mm ステップで移動させると新しいグリッドでは 1.25 mm、2.25 mm、3.25 mm の位置に移動します。
Add	このメニュー内のコマンドは、**Page Interface Editor** ウィンドウの図面エリアにアイテムを追加します。
Wire Port	**Entry** ウィンドウを表示します。. このウィンドウを使用して、バスポートを **Page Interface Editor** ウィンドウの図面エリアに追加します。 **EntryName** フィールドにポート名を入力して **OK** をクリックします。そして **Page Interface Editor** ウィンドウの図面エリア内にクリックしてポートを配置してください。
Bus Port	**Entry** ウィンドウを表示します。 このウィンドウを使用して、バスポートを **Page Interface Editor** ウィンドウの図面エリアに追加します。 **EntryName** フィールドにポート名を入れて **OK** をクリックします。そして **Page Interface Editor** ウィンドウの図面エリア内をクリックしてポートを配置してください。
Text/Vector Graphics	文字列とグラフィックツールのドロップダウンリストを表示します。 これらのツールを使用して文字列やベクトルグラフィックを **Page Interface Editor** ウィンドウの図面エリアに追加します。
Tools	このメニュー内のコマンドは距離（長さ）を測ります。
Measure	**Measure** ウィンドウを表示します。. このウィンドウはポインタをドラッグして決定した 2 つの点の距離を表示します。

Page Interface Editor ウィンドウツールバー



Page Interface Editor ウィンドウツールバーのボタンでよく使用される **Page Interface Editor** ウィンドウコマンドにアクセスできます。

メニューバーの全有効コマンドのリストは、*83*ページの***Page Interface Editor*** をご覧ください。

Page Interface Editor ウィンドウツールバー

Button		Description
	Print（印刷）	**Print** ウィンドウを表示します。 このウィンドウを使用して **Page Interface Editor** ウィンドウの図面エリアの内容を印刷します。
	Query/Change （クエリー/変更）	**Page Interface Editor** ウィンドウの図面エリア内で、アイテムのプロパティ（テキスト、テキストのサイズ、色、線幅など）を変更するときに使用します。 プロパティを変更したいアイテムをクリックしてください。選択したアイテムに適したダイアログボックスが表示されます。このウィンドウ内でプロパティを変更します。
	Move（移動）	**Page Interface Editor** ウィンドウの図面エリア内で選択したアイテムを移動します。
	Stretch （ストレッチ、伸縮）	選択した頂点を移動することで線や円弧などのグラフィックアイテムをストレッチ（伸縮）します。 選択したアイテムを移動します。
	Copy/Repeat （コピー/繰り返し）	**Page Interface Editor** ウィンドウの図面エリア内で、選択したアイテムをコピーします。選択したアイテムは白くなります。 クリックしてコピーしたアイテムを置きます。Esc を押すとコピーを中止します。 このコマンドは **Page Interface Editor** ウィンドウの図面エリア内でのみ有効です。
	Delete（削除）	**Page Interface Editor** ウィンドウの図面エリア内で、選択したアイテムを削除します。選択したアイテムは白くなります。 独立したアイテムや同一アイテムを選択すると **Attributes**（属性）ウィンドウが表示されます。 **Attributes** ウィンドウで **OK** をクリックして選択アイテムを削除します。 **Select Item Class** ウィンドウは、複数の異なるアイテムを選択するときに表示されます。 このウィンドウを使用して、削除したいアイテムを選択します。アスタリスク (*) で削除するアイテムを指定、ダッシュ (-) で削除しないアイテムを指定します。 **Select Item Class** ウィンドウで **OK** をクリックして選択アイテムを削除します。
	Undo （アンドゥ、取り消し）	プログラミング操作を元に戻します。 コンピュータのメモリに応じて最大 10 操作まで元に戻すことができます。

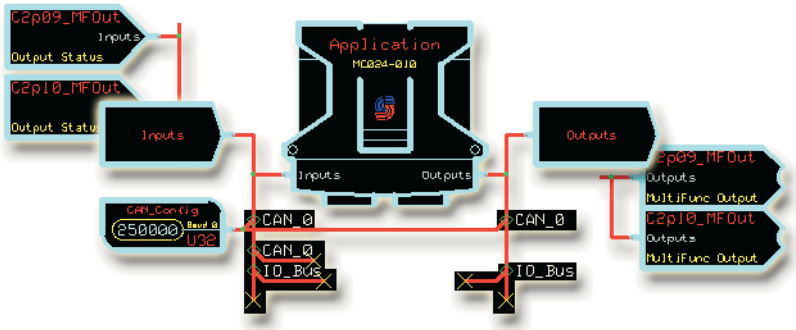
Page Interface Editor ウィンドウツールバー

Button		Description
	Redo（アンドゥの取り消し）	**Undo** 操作を元に戻します。 コンピュータのメモリに応じて、最大 10 操作まで **Undo** コマンドを元に戻すことができます。 **Options** ウィンドウによって Undo/Redo 機能を有効化/無効化することができます。このウィンドウを表示するには **Setup** メニューで **Options** をクリックします。
	Cut to Clipboard（クリップボードへカット）	**Page Interface Editor** ウィンドウの図面エリア内で選択したアイテムを削除し、それらをクリップボードにコピーします。
	Copy to Clipboard（クリップボードへコピー）	**Page Interface Editor** ウィンドウの図面エリア内で選択したアイテムをクリップボードにコピーします。
	Paste from Clipboard（クリップボードから貼り付け）	クリップボードの内容を **Page Interface Editor** ウィンドウの図面エリアにペーストします。
	Zoom In（ズームイン）	ズームインして、ポインタをクリックした場所を **Page Interface Editor** ウィンドウの図面エリアの中央に移動します。
	Zoom Out（ズームアウト）	ズームアウトして、ポインタをクリックした場所を **Page Interface Editor** ウィンドウの図面エリアの中央に移動します。
	Wire Port（ワイヤポート）	**Entry** ウィンドウを表示します。. このウィンドウを使用して、バスポートを **Page Interface Editor** ウィンドウの図面エリアに追加します。 **EntryName** フィールドにポート名を入力して **OK** をクリックします。そして **Page Interface Editor** ウィンドウの図面エリア内にクリックしてポートを配置してください。
	Bus Port（バスポート）	**Entry** ウィンドウを表示します。 このウィンドウを使用して、バスポートを **Page Interface Editor** ウィンドウの図面エリアに追加します。 **EntryName** フィールドにポート名を入れて **OK** をクリックします。そして **Page Interface Editor** ウィンドウの図面エリア内をクリックしてポートを配置してください。
	Thin Line（細線）	細い青線を描画します。通常、ページへの単独信号の入出力を表すのに使います。
	Thick Line（太線）	太い青線を描画します。通常、ページへのバスの入出力またはページ境界を表すのに使います。
	Hollow Rectangle（中空長方形）	太線で中空の長方形を描画します。
	Filled Rectangle（塗りつぶし長方形）	塗りつぶした長方形を描画します。
	Thin Arc（細い円弧）	細線で円弧を描画します。
	Thick Arc（太い円弧）	太線で円弧を描画します。
	Hollow Circle（中空円）	細線で中空の円を描画します。
	Filled Circle（塗りつぶし円）	塗りつぶした円を描画します。
ABC	Text（文字列）	**Page Interface Editor** ウィンドウの図面エリアをクリックした後、**Text Attributes** ウィンドウを表示します。 このウィンドウを使用して **Page Interface Editor** ウィンドウの図面エリアに文字列を追加します。

Page Interface Editor ウィンドウツールバー

Button		Description
	Working Grid（作業グリッド）	**Define Work Grid** ウィンドウを表示します。 このウィンドウを使用して作業グリッドのサイズ（グリッド線の間隔）を定義します。 **PLUS+1 GUIDE**、**Page Interface Editor**、**Module Viewer** ウィンドウの図面エリア内のアイテムはこの作業グリッドにスナップします。これは目に見えないグリッドです。
	Close（閉じる）	**Page Interface Editor** ウィンドウを閉じます。

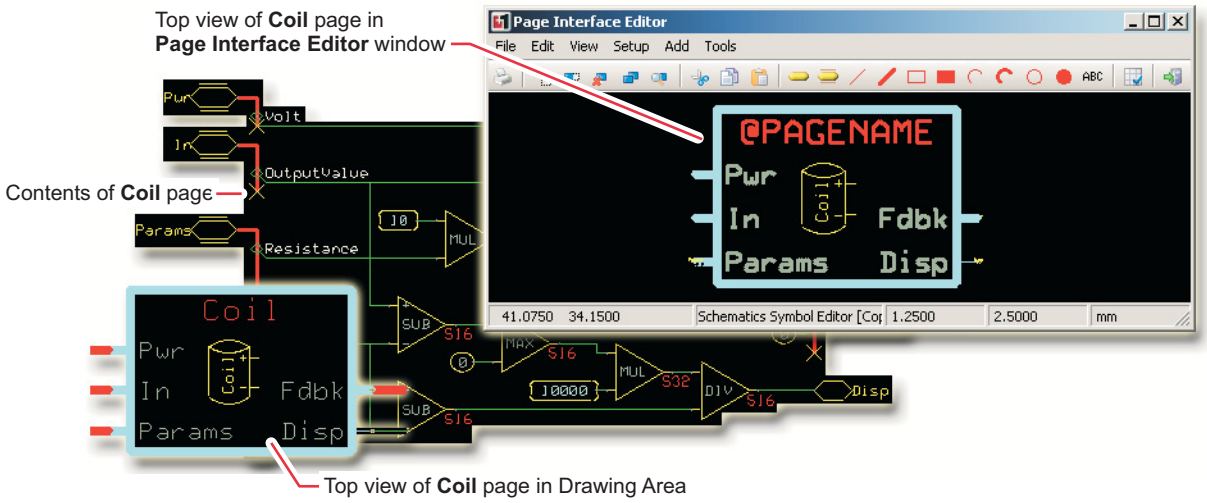
ページ、ページのトップビュー、ページインタフェースエディタのウィンドウについて



ザウアーダンフォス･ダイキンから提供されているアプリケーションテンプレートには **Inputs**、**Inputs**、**Application** などのページが含まれています。幾つかのページには内側に **MFIn**、**MFOut** などのページを含んでいます。これらのページによってアプリケーションテンプレートは論理的に構造化されています。

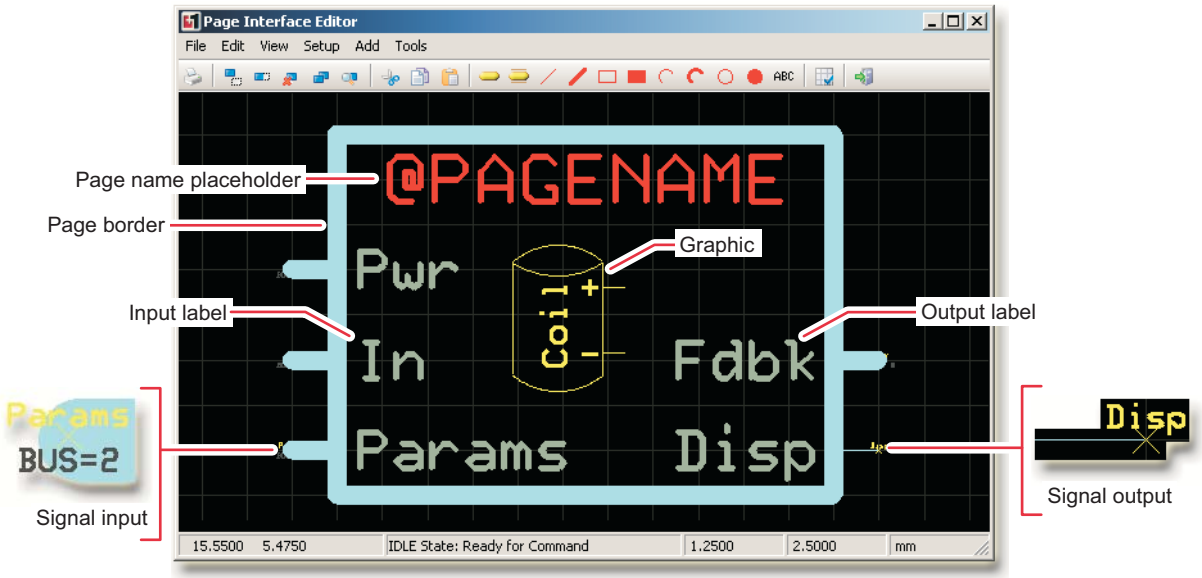
ページはさらに以下のような目的のために独自に作成することができます。

- コンポーネントをレイアウトするために必要なスペースとして
- アプリケーションの機能を論理的に構造化するため
- プログラムの乱雑や混乱を避けるため
- アプリケーション内、あるいは他のアプリケーションで使うための「パッケージ」を作成するため



ページのトップビューを編集することはページ作成の最初のステップです。ページのトップビューはページ内容の概観となり、ページに入るときにどう見えるかを決定するものです。

Page Interface Editor ウィンドウの図面エリア内のページのトップビューを編集します。



Page Interface Editor ウィンドウ内のツールを使って以下のことを行います

- ページ境界線の描画
- ページ名のプレースホルダ**@PAGENAME** の配置
- 信号入力と出力の追加
- 入力と出力のラベル付け
- グラフィックの描画

@PAGENAME プレースホルダの **Layer** 値は **PageName** のままにしておいてください。この値を変更するとページ名が消失することがあります。また、コンパイルエラーや PLUS+1 プログラムの応用機能に支障をきたす場合があります。

Page Interface Editor ウィンドウを表示して以下のページのトップビューを編集します

- **Add** メニューの **Page** コマンドで図面エリアに追加された新規ページ

  詳細な情報については**93**ページの*Page コマンドでページを追加/ページインタフェースエディタウィンドウの表示*を参照してください。

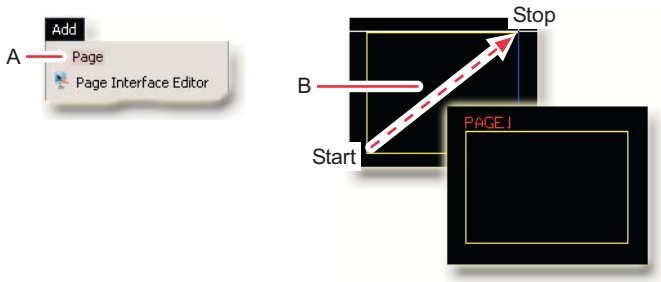
- **Components** タブから図面エリアにドラッグした **Basic Page**

  詳細な情報については**94**ページの*Basic Pageの追加/ページインタフェースエディタウィンドウの表示*を参照してください。.

- サイズ、ラベル、グラフィック、または入力や出力の変更が必要な古いページ

  詳細な情報については**95**ページの*古いページの変更/ページインタフェースエディタウィンドウの表示*を参照してください。

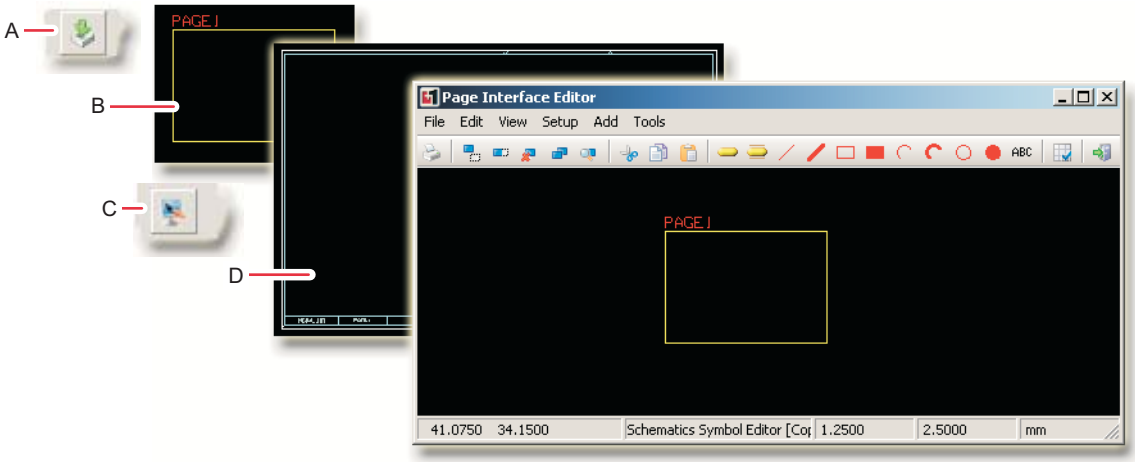
Page コマンドでページを追加/ページインタフェースエディタウィンドウの表示



1. 図面エリアにページを追加します

   A. **Add** メニュー内の **Page** をクリックします

   B. 図面エリア内で左下から右上に向かってクリック&ドラッグして新しいページを追加します



2. **Page Interface Editor** ウィンドウを表示します。

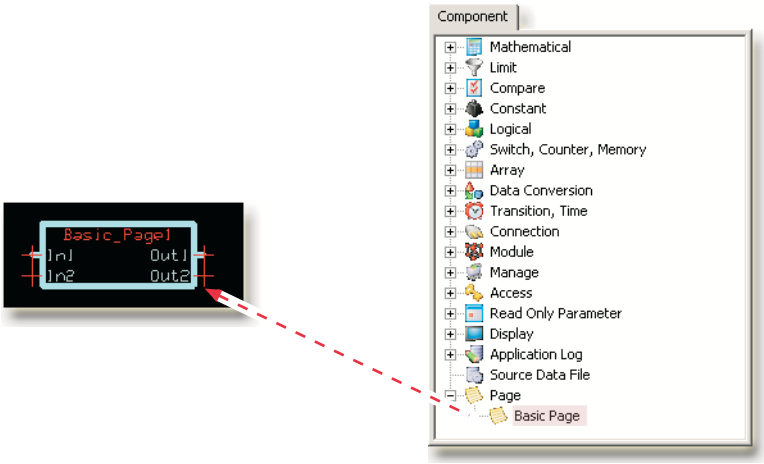
   A. ツールバーの Enter page ボタンを押します

   B. ページに入ります

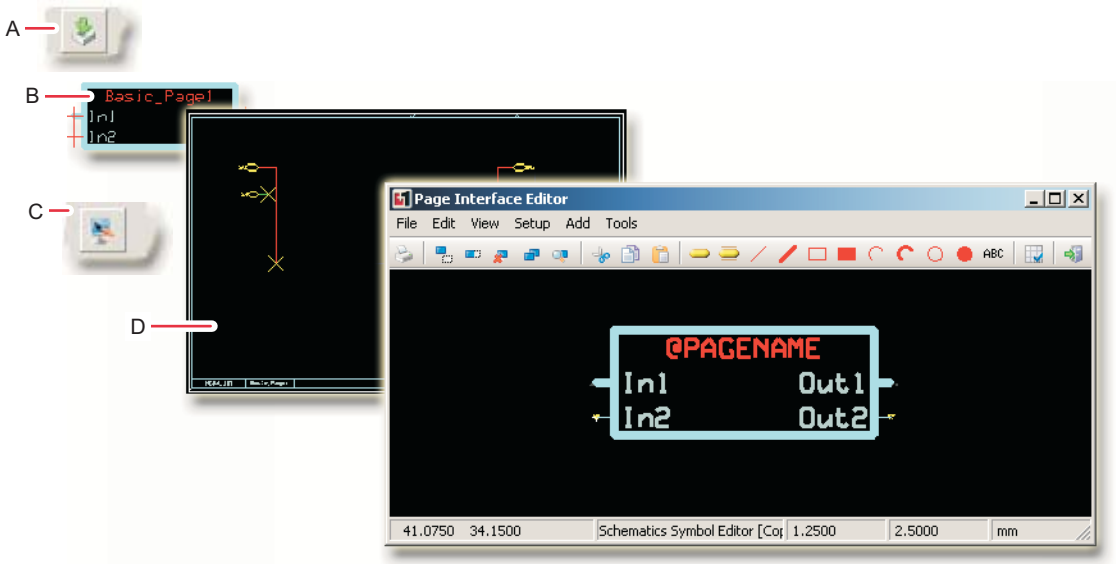
   C. ツールバーの Page Interface Editor ボタンを押します

   図面エリア内に **Page Interface Editor** ウィンドウが新しいページのトップビューを表示します。

Basic Pageの追加/ページインタフェースエディタウィンドウの表示



1. **Components** タブから **Basic Page** を図面エリアにドラッグします。



2. **Page Interface Editor** ウィンドウを表示します。

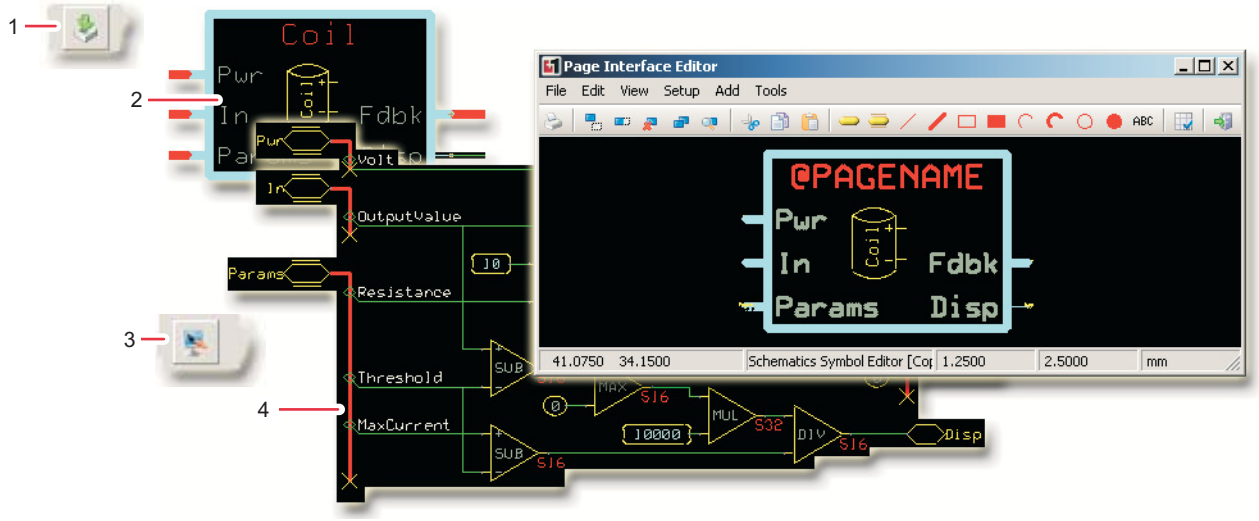
   A. ツールバーの Enter page ボタンをクリックします。

   B. **Basic Page** に入ります。

   C. ツールバーの Page Interface Editor ボタンをクリックします。

   図面エリア内に **Page Interface Editor** ウィンドウが **Basic Page** のトップビューを表示します。

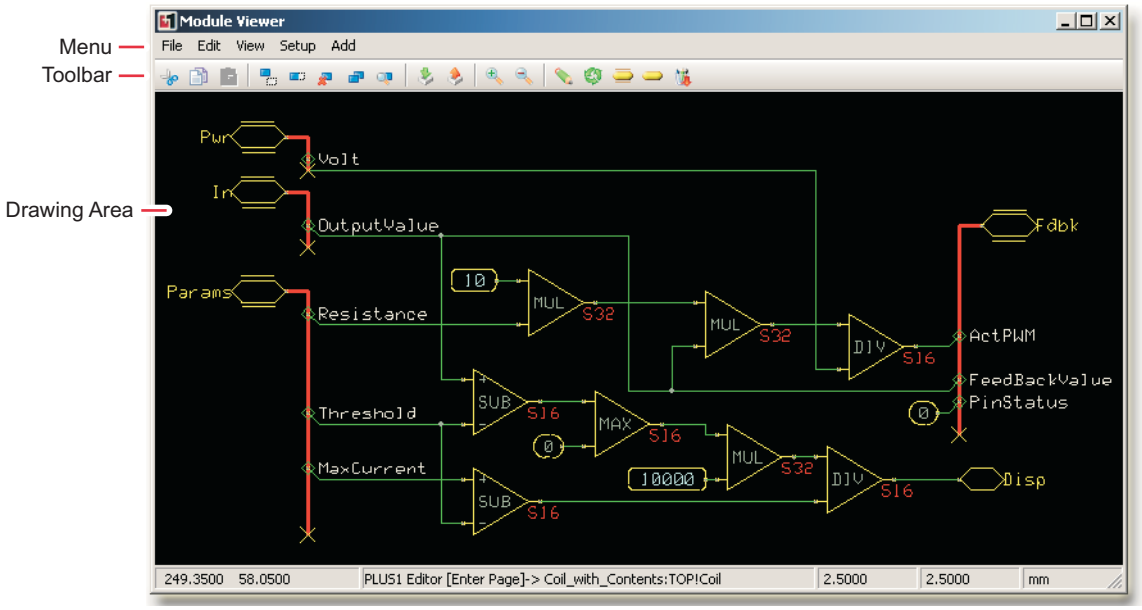
古いページの変更/ページインタフェースエディタウィンドウの表示



1. ツールバーの Enter page ボタンをクリックします。
2. トップビューを変更しようとする古いページに入ります.
3. ツールバーの Page Interface Editor ボタンをクリックします。

   図面エリア内に **Page Interface Editor** ウィンドウが古いページのトップビューを表示します。

Module Viewer ウィンドウ



Module Viewer ウィンドウはメインの **PLUS+1 GUIDE** ウィンドウとは別の、SCS ファイルを編集できるウィンドウを提供します。

このウィンドウは以下のような使い方が一般的です。Typical uses of this window include:

- 同一プロジェクト内で複数のページを見る

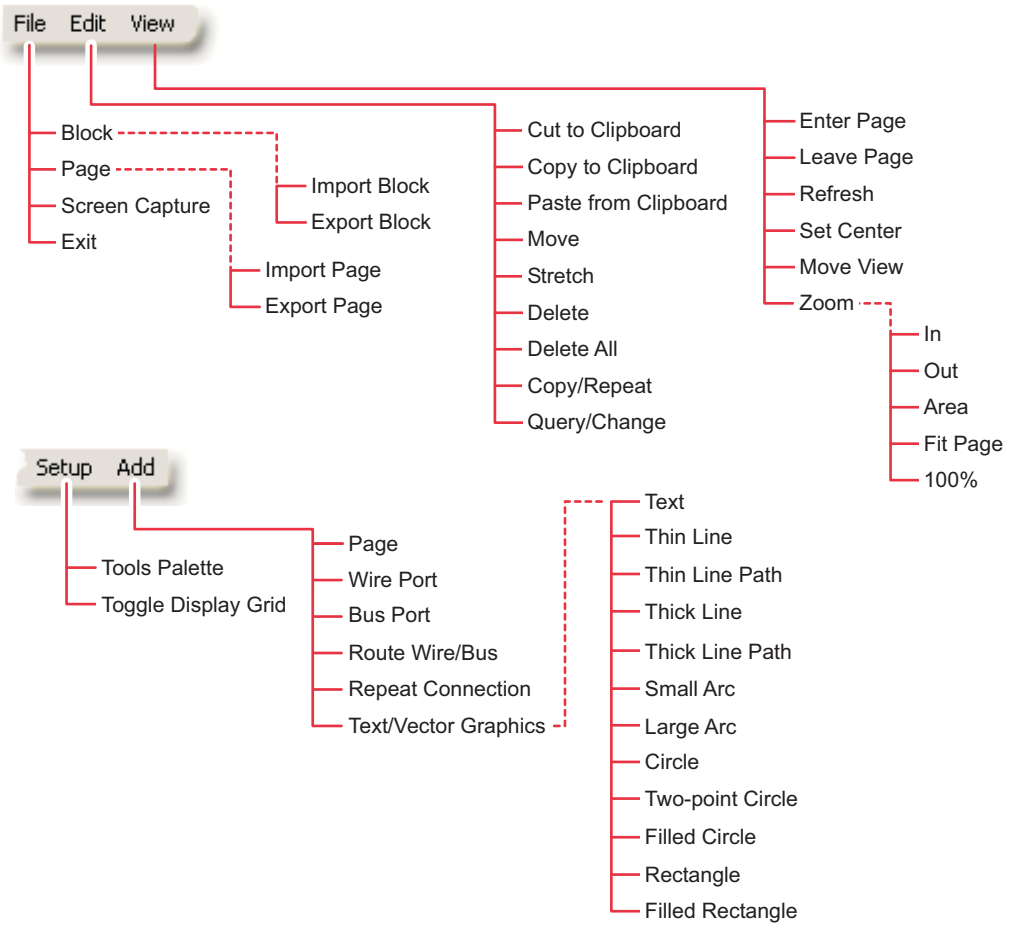
  （このウィンドウで現在のプロジェクトの SCS ファイルを開くことができます。）

- 二つのコントローラ間で CAN 通信の設定を連携させる

Module Viewer ウィンドウの要素

項目	説明
Menus（メニュー）	メニューバーを使用して **Module Viewer** ウィンドウコマンドにアクセスする
Toolbar（ツールバー）	ツールバーを使用してよく使われる **Module Viewer** ウィンドウコマンドにアクセスする
Drawing Area（図面エリア）	モジュールをここで編集する

Module Viewerウィンドウメニュー



メニューバーを使用して **Module Viewer** ウィンドウコマンドにアクセスします。

Module Viewerウィンドウツールバーはよく使われる**Module Viewer**コマンドを含みます。ツールバーについての詳細は***101***ページの***Module Viewer***を参照してください。

Module Viewer ウィンドウメニュー

Item	Description
File	このメニュー内のコマンドで SCS ファイルを管理します。
Block	**Import Block: PLUS1 Editor Load Block** ウィンドウを表示します。. このウィンドウを使用して **Module Viewer** ウィンドウの図面エリアにインポートする SCS ファイルを選択します。 同一ブロックを二つ以上インポートしたときはリンク機能によりブロックの内容がリンクされます。詳細な情報については *472* ページ、*リンクページについて*を参照してください。 **Export Block: Module Viewer** ウィンドウの図面エリア内でエクスポートするアイテムを選択した後、**Symbol Block Export Binary** ウィンドウを表示します。 このウィンドウを使用して選択したアイテムを SCS ファイルにエクスポートします。 **Import Block** コマンドと **Export Block** コマンドは **Define Areas** ブロックと **Define Screen** ブロックには使用できません。
Page	**Import Page: Module Viewer** ウィンドウの図面エリア内でページを選択した後 **Select Job File Name** ウィンドウを表示します。 このウィンドウを使用して SCS ファイルを選択し、その内容を選択したページにインポートします。 **Export Page: Module Viewer** ウィンドウの図面エリア内でページを選択した後 **Select/Define New Job File Name** ウィンドウを表示します。ページをインポートするときに SCS ファイル名がページ名として使用されます。 ページ名は 16 文字の制限があるため、SCS ファイル名も 16 文字に制限されます。長いページ名はコンパイルエラーの原因となります。 **Import Block** コマンドと **Export Block** コマンドは **Define Areas** ブロックと **Define Screen** ブロックには使用できません。
Screen Capture	**Module Viewer** ウィンドウの図面エリア内でアイテムを選択した後 **Select Screen Dump Format** ウィンドウを表示します。 このウィンドウを使用して選択したアイテムの印刷、***BMP*** や ***TIFF*** フォーマットのファイルへの保存、あるいは BMP や Metafile フォーマットとしてクリップボードにコピーを行います。
Close	**Module Viewer** ウィンドウを閉じます。
Edit	このメニュー内のコマンドで **Module Viewer** ウィンドウの図面エリア内のアイテムの変更や修正をします。
Cut to Clipboard	**Module Viewer** ウィンドウの図面エリア内で選択したアイテムを削除し、クリップボードにコピーします。
Copy to Clipboard	**Module Viewer** ウィンドウの図面エリア内で選択したアイテムをクリップボードにコピーします。
Paste from Clipboard	クリップボードの内容を **Module Viewer** ウィンドウの図面エリア内にペーストします。 リンク機能によりクリップボードからコピーされた複数のページは自動的にリンクされます。詳細な情報については *472* ページ、*リンクページについて*を参照してください。
Move	選択したアイテムを **Module Viewer** ウィンドウの図面エリア内に移動します。
Stretch	選択したルート上のセグメントをストレッチ（伸縮）したり、選択したアイテムを移動したりします。 ルートをクリックして頂点をルート上に追加します。追加された頂点は、ルート上の他の頂点の位置を変えずに移動させることができます。
Delete	**Module Viewer** ウィンドウの図面エリア内で選択したアイテムを削除します。選択したアイテムは白くなります。 単一のアイテムや同一アイテムを選択すると、**Attributes** ウィンドウが表示されます。 **Attributes** ウィンドウ内で **OK** をクリックして選択アイテムを削除します。 **Select Item Class** ウィンドウは、複数の異なるアイテムを選択するときに表示されます。 このウィンドウを用いて、削除したいアイテムを選択します。アスタリスク(*)で削除するアイテムを指定し、ダッシュ(-)で削除しないアイテムを指定します。 **Select Item Class** ウィンドウ内で **OK** をクリックして選択アイテムを削除します。

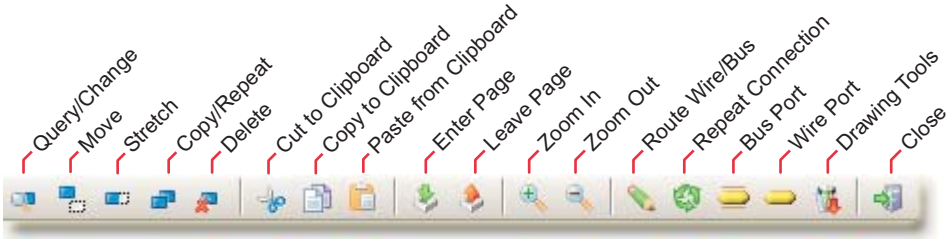
Module Viewer ウィンドウメニュー

Item	Description
Delete All	**Delete all items?**メッセージを含む **Question** ウィンドウを表示します。 **Yes** をクリックして **Module Viewer** ウィンドウの図面エリア内の全てのアイテムを削除します。
Copy/Repeat	**Module Viewer** ウィンドウの図面エリア内で選択したアイテムをコピーします。選択されたアイテムは白くなります。 クリックするとコピーしたアイテムの位置を確定します。Esc でコピーしたアイテムをキャンセルします。 このコマンドは図面エリア内でのみ有効です。 リンク機能により、コピーされたページは自動的にリンクされます。詳細な情報については *472* ページ、*リンクページについて*を参照してください。
Query/Change	**Module Viewer** ウィンドウの図面エリア内で、アイテムのプロパティ（データ型、テキスト、機能など）を変更するときに使用します。 プロパティを変更したいアイテムをクリックしてください。選択したアイテムに適したダイアログボックスが表示されます。このウィンドウ内でプロパティを変更します。
View	このメニュー内のコマンドは、**Module Viewer** ウィンドウの図面エリアのビューを変更します。
Enter Page	選択したページに入ります。 ページに入るには、ページ境界の内側をクリックするか、ページのポートをドラッグします。
Leave Page	現在のページから出ます。
Refresh	**Module Viewer** ウィンドウの図面エリアを再描画します。
Set Center	ポインタでクリックした場所を **Module Viewer** ウィンドウの図面エリアの中央に移動します。
Move View	白い境界線を持つ移動可能で透明な長方形を表示します。クリックすると **Module Viewer** ウィンドウの図面エリアがこの長方形の境界に入るように中心が移動します。
Zoom	**In**： ズームインして、ポインタをクリックした場所を **Module Viewer** ウィンドウの図面エリアの中央に移動します。 **Out**： ズームアウトして、ポインタをクリックした場所を **Module Viewer** ウィンドウの図面エリアの中央に移動します。 **Area**：**Module Viewer** ウィンドウの図面エリアを 2 箇所クリックしたポイントで決まる範囲にズームします。 **Fit Page**： ページ内の全てのアイテムが **Module Viewer** ウィンドウの図面エリア内にフィットするようにビューのサイズを変更します。 **100%**：**Module Viewer** ウィンドウの図面エリアを最小ズームビューと最大ズームビューの中間サイズにズームします。
Setup	このメニュー内のコマンドで **Module Viewer** ウィンドウの作業環境を設定します。
Toggle Display Grid	ディスプレイグリッドをオン/オフします。 ディスプレイグリッドとは **PLUS+1 GUIDE**、**Page Interface Editor** および **Module Viewer** ウィンドウの図面エリアで表示されるグリッドです。The display grid is the grid that you see in the Drawing Areas of the **PLUS+1 GUIDE**, **Page Interface Editor**, and **Module Viewer** windows.
Tools Palette	**Icon Menu** ウィンドウを表示します。
Add	このメニューのコマンドは、アイテムを **Module Viewer** の図面エリアに追加します。
Page	**Module Viewer** ウィンドウの図面エリア内でクリック&ドラッグした場所にページを追加します。
Wire Port	**Select Signal–Schematics Design** ウィンドウを表示します。 このウィンドウを使用して **Module Viewer** ウィンドウの図面エリアにワイヤポートを追加します。 **EntryName** フィールドにポート名を入力し **OK** を押してください。そして **Module Viewer** ウィンドウの図面エリア内をクリックしてポートを配置します。
Bus Port	**Select Signal–Schematics Design** ウィンドウを表示します。 このウィンドウを使用して **Module Viewer** ウィンドウの図面エリアにバスポートを追加します。 **EntryName** フィールドにポート名を入力し **OK** を押してください。そして **Module Viewer** ウィンドウの図面エリア内をクリックしてポートを配置します。

Module Viewer ウィンドウメニュー

Item	Description
Route Wire/Bus	**Module Viewer** ウィンドウの図面エリアのクリックした位置から緑色の単一信号ワイヤまたは赤色のマルチ信号バスのいずれかの配線を始めます。 バスポートから配線を始めるとバスになります。ワイヤポートから配線を始めるとワイヤになります。 K でワイヤとバスのルートをトグルで切り換えます。 F9 で接続されていないルートを終端します。
Repeat Connection	**Module Viewer** ウィンドウの図面エリア内で、選択したルート接続を繰り返します。 繰り返し作成された接続は自動的に元の名前に数字が追加されます。***Signal_Name***、***Signal_Name2***、***Signal_Name3***、***Signal_Name4*** など。
Text/Vector Graphics	テキストとグラフィック用コマンドのメニューを表示します。 このコマンドを使用してテキストやグラフィックエレメントを **Module Viewer** ウィンドウの図面エリアに追加します。

Module Viewerウィンドウツールバー



Module Viewer ウィンドウのツールバーのボタンは、よく使用する **Module Viewer** コマンドにアクセスします。

メニューバーでアクセス可能な全コマンドのリストは*97*ページの*Module Viewer*を参照してください。

Module Viewer ウィンドウツールバー

Button		Description
	Query/Change	**Module Viewer** ウィンドウの図面エリア内のアイテムのプロパティ（データ型、文字、機能など）を変更するときに使用します。 プロパティを変更したいアイテムをクリックしてください。選択したアイテムに適したダイアログボックスが表示されます。このウィンドウ内でプロパティを変更してください。
	Move	**Module Viewer** ウィンドウの図面エリア内で、選択したアイテムを移動します。
	Stretch	選択したルートのセグメントをストレッチし、選択したアイテムを移動します。 ルートをクリックしてルートに頂点を追加します。新しい頂点はルート上の他の頂点の位置を変えずに移動させることができます。
	Copy/Repeat	**Module Viewer** ウィンドウの図面エリア内で、選択したアイテムをコピーします。選択されたアイテムは白くなります。クリックするとコピーしたアイテムの位置を確定します。Esc を押してコピーしたアイテムをキャンセルします。 このコマンドは現在の図面エリア内でのみ有効です。 リンク機能によりコピーしたページを自動的にリンクします。詳細な情報については *472* ページ、*リンクページについて*を参照してください。
	Delete	**Module Viewer** ウィンドウの図面エリアで、選択したアイテムを削除します。選択したアイテムは白くなります。 独立したアイテムや同一アイテムを選択すると、**Attributes**（属性）ウインドウが表示されます。**OK** をクリックして選択アイテムを削除します。 **Select Item Class** ウインドウは、複数の異なるアイテムを選択するときに表示されます。このウインドウを使用して、削除したいアイテムを選択します。 アスタリスク(*)で削除されるアイテムを指定します。 ダッシュ(-)で削除しないアイテムを指定します。 **Select Item Class** ウィンドウで **OK** をクリックして選択アイテムを削除します。
	Cut to Clipboard	**Module Viewer** ウィンドウの図面エリア内で、選択したアイテムを削除してクリップボードにコピーします。
	Copy to Clipboard	**Module Viewer** ウィンドウの図面エリア内で、選択したアイテムをクリップボードにコピーします。

Module Viewer ウィンドウツールバー

Button		Description
	Paste from Clipboard	**Module Viewer** ウィンドウの図面エリア内で、クリップボードの内容を貼り付けます。 リンク機能により、クリップボードからコピーされた複製ページの内容は自動的にリンクされます。詳細な情報については *472* ページ、*リンクページについて*を参照してください。
	Enter Page	選択したページに入ります。 ページに入るには、ページ境界の内側をクリックするか、ページのポートをドラッグします。
	Leave Page	現在のページから出ます。
	Zoom In	ズームインして、ポインタをクリックした場所を **Module Viewer** ウィンドウの図面エリアの中央に移動します。
	Zoom Out	ズームアウトして、ポインタをクリックした場所を **Module Viewer** ウィンドウの図面エリアの中央に移動します。
	Route Wire/Bus	**Module Viewer** ウィンドウの図面エリアのクリックした位置から緑色の単一信号ワイヤまたは赤色のマルチ信号バスのいずれかの配線を始めます。 バスポートから配線を始めるとバスになります。ワイヤポートから配線を始めるとワイヤになります。 K でワイヤとバスのルートをトグルで切り換えます。 F9 で接続されていないルートを終端します。
	Repeat Connection	**Module Viewer** ウィンドウの図面エリア内で、選択したルート接続を複製します。 繰り返された接続は自動的に元の名前に番号が追加されます。*Signal_Name*、*Signal_Name2*、*Signal_Name3*、*Signal_Name4* などです。
	Bus Port	**Select Signal–Schematics Design** ウィンドウを表示します。 このウィンドウを使用して **Module Viewer** ウィンドウの図面エリアにバスポートを追加します。 ポート名を **EntryName** フィールドに入力して **OK** をクリックしてください。そして **Module Viewer** ウィンドウの図面エリア内をクリックしてポートを配置します。
	Wire Port	**Select Signal–Schematics Design** ウィンドウを表示します。 このウィンドウを使用して **Module Viewer** ウィンドウの図面エリアにワイヤポートを追加します。 ポート名を **EntryName** フィールドに入力して **OK** をクリックしてください。そして **Module Viewer** ウィンドウの図面エリア内をクリックしてポートを配置します。
	Drawing Tools	テキストと基本グラフィックツールのドロップダウンリストを表示します。 このコマンドを使用してテキストやベクトルグラフィック要素を **Module Viewer** ウィンドウの図面エリアに追加します。
	Close	**Module Viewer** を閉じます。.

データ型

本章では、以下の説明を行います。

- データ型

  各データ型は、値の範囲を示すものと、特定機能を示すものとがあります。
- 値の範囲が狭すぎるデータ型を使用した結果生じるオーバーフロー状態
- タイムベースのデータ型の使用方法
- アレイの基本特徴

この章の内容

データ型.....103
データ型 104
オーバーフロー条件について.....105
Time Base データ型について.....106
Array（配列）データ型について 108

データ型

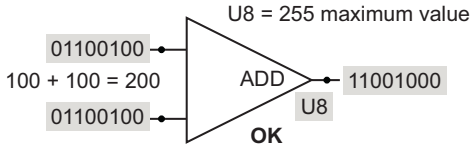
以下の表は、PLUS+1 GUIDE アプリケーションで使用するデータ型をリストアップしています。

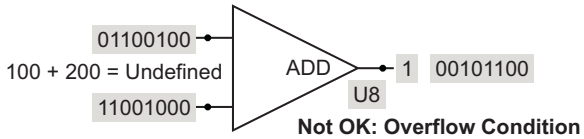
データ型

データ型	説明	範囲
BOOL	True または False	T-F
U8	符号なし (Unsigned) 8 ビット	0 ～+255
S8	符号つき (Signed) 8 ビット	-128 ～ +127
U16	符号なし 16 ビット	0 ～ +65535
S16	符号つき 16 ビット	-32768 ～ +32767
U32	符号なし 32 ビット	0 ～ +4294967295
S32	符号つき 32 ビット	-2147483648 ～ +2147483647
TL	テキストラベル (Text Label) ：テキストテーブル内文字列へのポインタ	
PXO	ピクセルオブジェクト (PiXel Object) ：外部グラフィックへのポインタ	
PORT	ハードウェアポート：ハードウェアポート名と適合させること	
T	タイムベース	
FILE	グラフィックスを表示するウインドウを定義	
FONT	使用フォント：OS によって定義	
COLOR	使用カラー：OS によって定義	
OBN	オブジェクト名 (OBject Name)	
NULL	結合なし	
LANG	表示言語を設定	
ARRAY	最大 32767 エレメントまで可能	

オーバーフロー条件について

オーバーフローは出力値が出力に割り当てられたデータ型よりも大きくなったときに起こります。



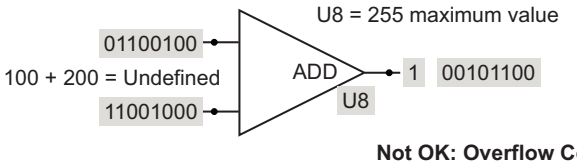


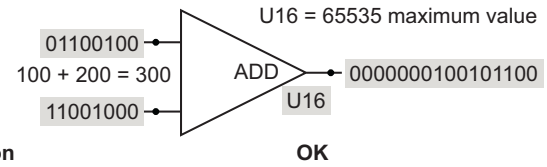
上の図の場合

- **Add** コンポーネントは 2 つの U8 値を追加し、結果の U8 値を出力します。
- データ型 U8 は 8 ビットで、最大値 255 です。
  - OK: 100 + 100 = 200:出力は 255 未満です。
  - Not OK: 100 + 200 = 300:出力が 255 を超えたので、オーバーフローが生じます。

    オーバーフローによって 44 などの誤った出力値を出します。

整数の出力にデータを割り当てるとき、オーバーフローしないように、各データ型の最大可能値を念頭においてください。





上の図の場合

- Not OK: 100 + 200 = 300:出力が 255 を超えたので、オーバーフローが生じます。
- OK: 100 + 200 = 300:U16 データの最大値は 65535 なので、オーバーフローではありません。

Add Clamped(クランプ加算)および **Subtract Clamped**(クランプ減算)などの "Clamped" (クランプ)コンポーネントは、以下のようにして、オーバーフロー状態を扱います。

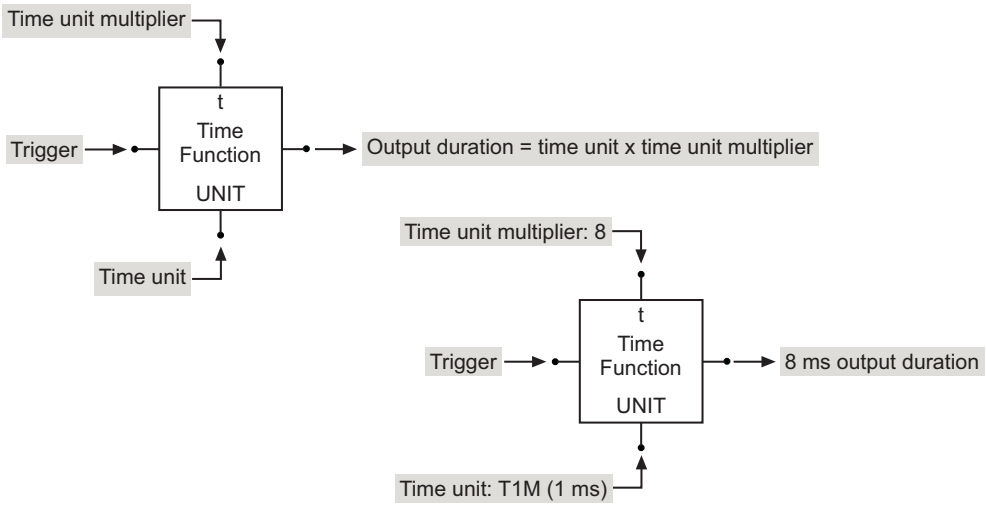
- 出力データ型によって可能な最大値や最小値で出力値をクランプする。
- オーバーフロー状態を起こす各プログラムループの中でブール出力を T に設定する。

Time Baseデータ型について

Time Base データ型は **Time and Transition**（時間と遷移）コンポーネントと共に使用します。

Time and Transition コンポーネントの 3 例を以下に示します。

- **On Delay**（オン・ディレイ）：F から T へのブール信号を指定時間だけ遅らせます。
- **Oscillator**（オシレータ）： 指定の周波数で変動するブール信号を出力します。
- **Pulse**（パルス）：指定時間ブール出力を T に保持し、その後 F に戻ります。



上図は、時間単位が以下のように使用されることを示します。

- 定数として時間に関するコンポーネントに適用する。
- 時間単位乗数 (Time unit multiplier) で掛算を行い、出力継続時間を設定する。

以下の表は PLUS+1 GUIDE の時間単位とそれが示す値をリストアップしています。ハードウェアの選択によって選択できる時間単位が決まります。

タイムベースの値

タイムベースの値	タイムベースの略記法
1 ms	T1M
10 ms	T10M
100 ms	T100M
1 s	T1S
60 s	T60S
1 h	TIH
ループ時間	TLOOP*

*ループ時間またはプログラムループ： 出力の変更を行うために入力の変更にかかる時間。

分解能

処理時間、出力継続時間、時間単位は、分解能（変動値）を最小出力継続時間と最大出力継続時間の間に設定します。

（最小出力継続時間）=（出力継続時間）−（時間単位）
（最大出力継続時間）=（出力継続時間）+（処理時間）
但し、
（出力継続時間）=（時間単位）×（時間単位乗数）
（処理時間）=（入力の変化が出力を変化させるのに掛かる時間）

- 例 1：

  100 ms の時間単位の場合、時間単位乗数は 5、処理時間は 15 ms

  （最小出力継続時間） = 400 *ms*
  （最小出力継続時間）=（出力継続時間）−（時間単位）
  （最小出力継続時間）=（100 *ms* × 5）− 100 *ms*

  （最大出力継続時間） = 515 *ms*
  （最大出力継続時間）=（出力継続時間）+（処理時間）
  （最大出力継続時間）=（100 *ms* × 5）+ 15 *ms*

- 例 2：

  10 ms の時間単位の場合、時間単位乗数 25、処理時間は 15 ms

  （最小出力継続時間） = 240 *ms*
  （最小出力継続時間）=（出力継続時間）−（時間単位）
  （最小出力継続時間）=（10 *ms* × 25）− 10 *ms*

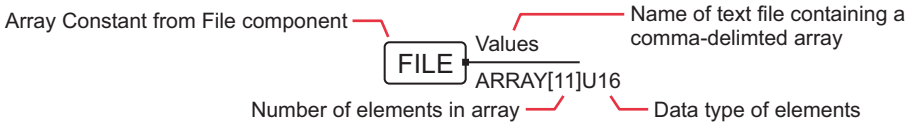
  （最大出力継続時間） = 265 *ms*
  （最大出力継続時間）=（出力継続時間）+（処理時間）
  （最大出力継続時間）=（10 *ms* × 25）+ 15 *ms*

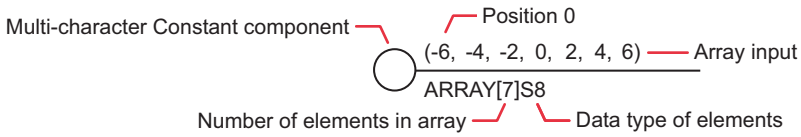
TLOOP は、その時間単位値が処理時間なので、常に出力継続時間が同じです。

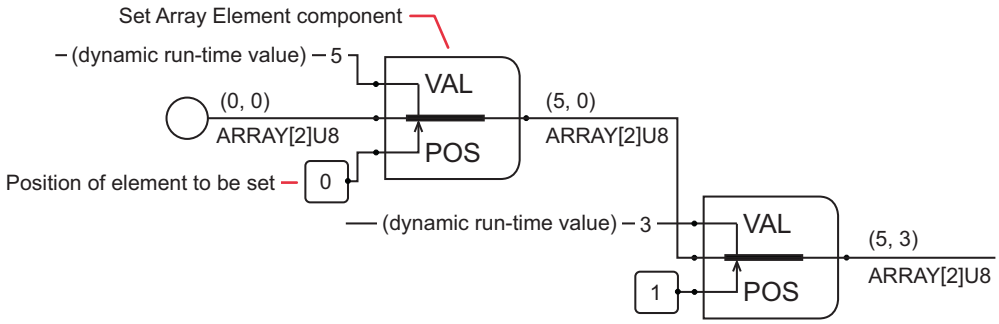
Array（配列）データ型について

- 配列は、最大 32,767 まで要素を持つことが可能。
- 配列の最初の要素は、ポジション 0。
- マルチキャラクタ定数コンポーネントから配列入力を定義する場合
  - 配列を括弧（）内に括る。
  - 配列内の各要素の分離にはコンマを使用する。
- テキストファイルからの配列入力を定義する場合、テキスト内でコンマを使用して、配列内の各要素を分離します。
- 配列内のデータ型を定義する場合
  - 配列内の要素数を括弧［］内に入れる。
  - S8、U16、BOOL のような配列のデータ型を選択する。

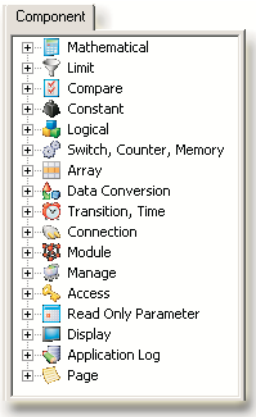
以下の図は、配列および配列関連コンポーネントの例を示しています。







コンポーネント



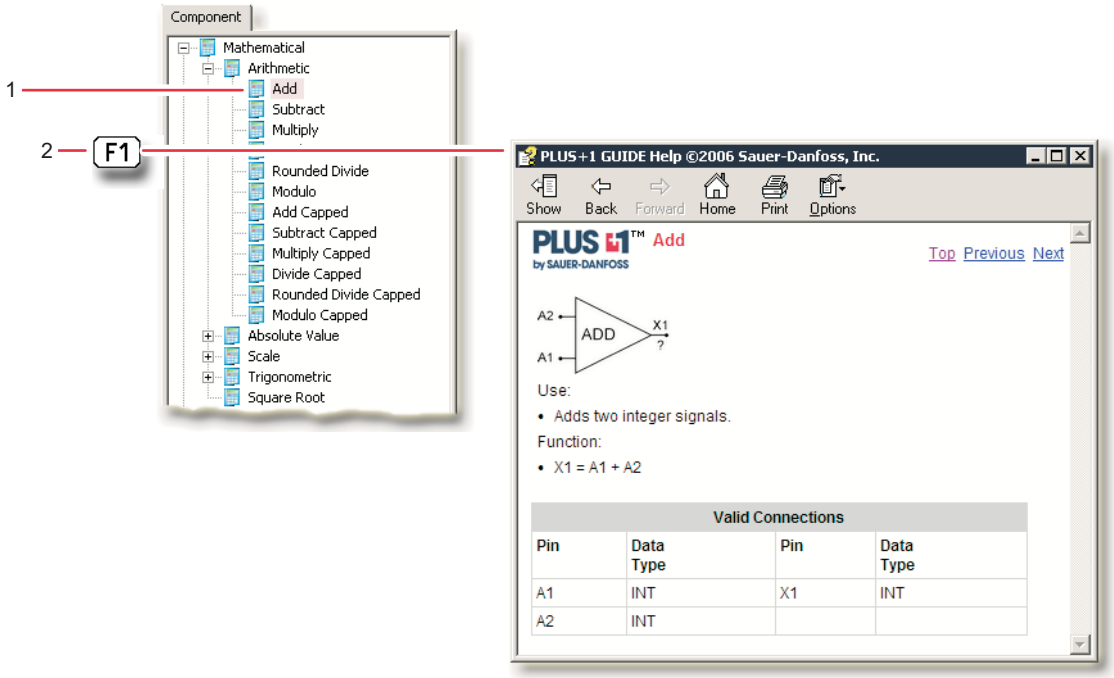
本章では、**Selector**（セレクタ）ウインドウの **Component**（コンポーネント）タブのコンポーネントについて説明します。

この章の内容

コンポーネント 109
コンポーネントに関するコンテクストヘルプ画面 110
コンポーネント記述について 111
Mathematical (数学的).....113
Limit（リミット）.....140
Compare（比較）.....150
Constant（定数）.....165
Logical（論理）.....175
Switch, Counter, Memory（スイッチ、カウンタ、メモリ）.....192
Array（配列）.....217
Data Conversion（データ変換）.....234
Transition, Time（遷移、時間）.....245
Connection（接続）.....263
Non-Volatile Memory Input（不揮発性メモリ入力）.....298
Module（モジュール）.....300
Manage（管理）.....309
Access（アクセス）.....320
Read-only Parameter（読取専用パラメータ）.....334
Display（ディスプレイ）.....341
Application Log（アプリケーションログ）.....364
Page（ページ）.....368

コンポーネントに関するコンテクストヘルプ画面

PLUS+1 GUIDE プログラムでは、すべてのコンポーネントに関して、コンテクストに応じたヘルプ画面を表示できます。



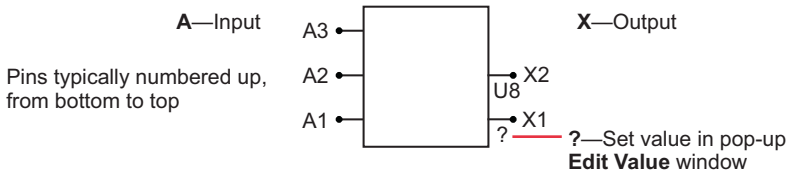
コンポーネントに関するコンテクストに応じたヘルプ画面を表示するには

1. ヘルプを必要とするコンポーネントをクリックします。
2. F1 を押します。

コンポーネント記述について

本章では、図、説明文、表を用いて、コンポーネントの説明を行います。

典型的なコンポーネント図



典型的なコンポーネント図において

- **A1**、**A2**、**A3** 等の **A** ラベルは、入力ピンを示しています。
- **X1**、**X2** 等の **X** ラベルは、出力ピンを示しています。
- **?** は以下の場合に選択しなくてはならない値を示しています。
  - データ型が選択されていない。
  - デフォルトのデータ型を変更する必要がある。

  ポップアップ **Edit Value** ウインドウまたは **Inspector**（インスペクタ）タブを用いて、データ型を変更します。

典型的な有効接続表

Valid connections for **A** input pins

Valid connections

Pin	Type	Pin	Type
A1	BOOL	X1	BOOL
A2	INT	X2	U16
	TIME		

Valid connections for **X** output pins

各コンポーネントには、「有効接続」表があります。この表によって、コンポーネントピンに接続できるデータ型を識別できます。

この表には、**BOOL** や **U16** 等の単一データ型または **INT**（整数）等のデータ型グループが示されます。

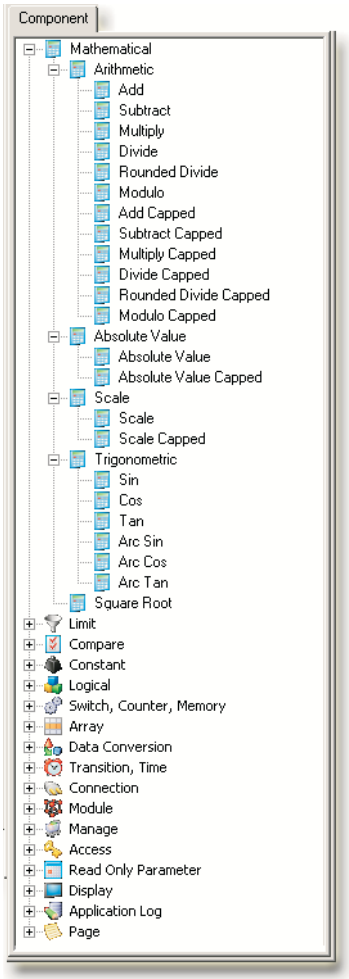
表にデータ型グループが示されている場合、そのグループ内のデータ型のいずれでもピンに接続できます。例えば、**INT** が示されていれば、**U8**、**U16**、**U32**、**S8**、**S16**、**S32** データ型をピンに接続できます。

以下の表に、各データ型グループ内のデータ型が示されています。

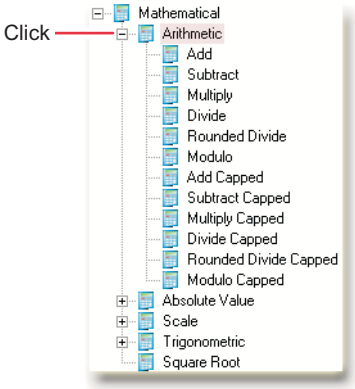
データ型グループ

データ型グループ	データ型グループ内の個別データ型						
	BOOL	U8	U16	U32	S8	S16	S32
ARRAY（配列）	○	○	○	○	○	○	○
INT（整数）		○	○	○	○	○	○
UINT（符号なし整数）		○	○	○			
ALL(すべて)	すべての PLUS+1 データ型						

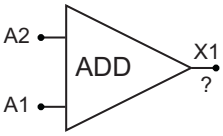
Mathematical (数学的)



Arithmetic（算術演算）



Add（加算）



用途

- 2つの整数信号を加算します。

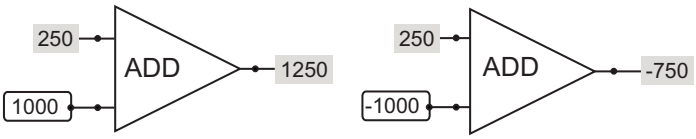
機能

- X1 = A1 + A2

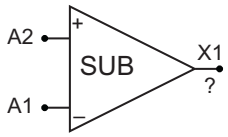
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例: Add



Subtract（減算）



用途

- 2つの整数信号を減算します。

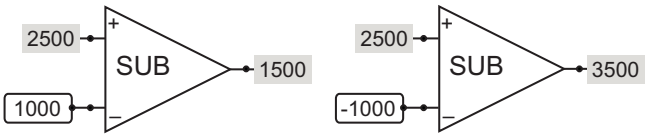
機能

- X1 = A2 – A1

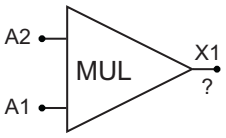
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例: Subtract



Multiply（乗算）



用途

- 2つの整数信号を乗算します。

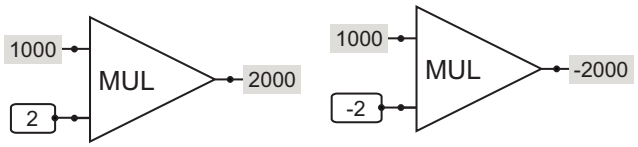
機能

- X1 = A1 ∗ A2

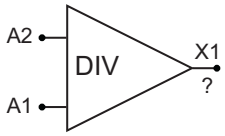
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例



Divide（除算）



用途

- 2つの整数信号を除算します。

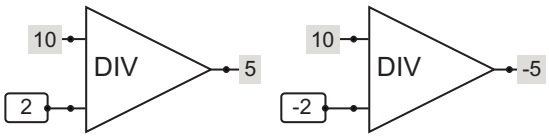
機能

- X1 = A2 ÷ A1

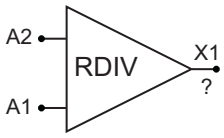
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例



Rounded Divide（丸め除算）



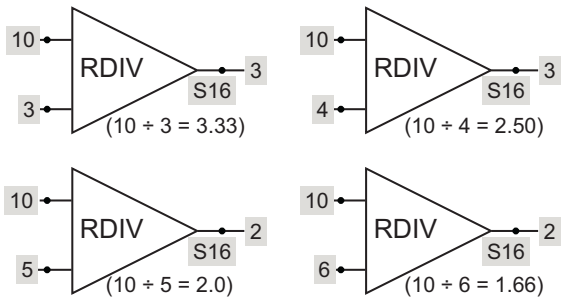
用途

- 2つの整数信号を除算し、丸めた計算結果を出力します。
- 機能
  - X1 = A2 ÷ A1 の出力を丸めます
    - 剰余が 0.5 以上の場合には切り上げ
    - 剰余が 0.5 未満の場合には切り下げ

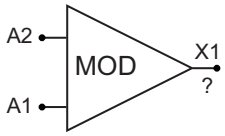
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例



Modulo（剰余）



- 2つの整数信号の除算から生じるモジュロ（剰余）を出力します。

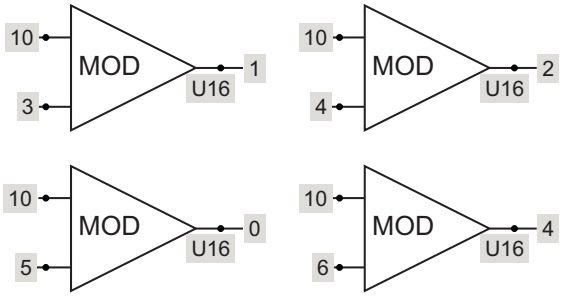
機能

- $X1 = A2 - \left(\left[\frac{A2}{A1} \right] \times A1 \right)$

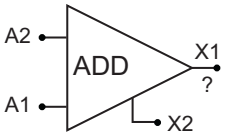
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例



Add Capped（上限付き加算）



用途

- 2つの整数信号を加算します。
- オーバーフローする場合には出力をクランプし、X2 出力を T（True）に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

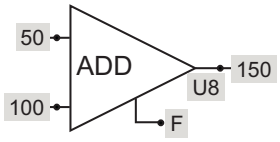
機能

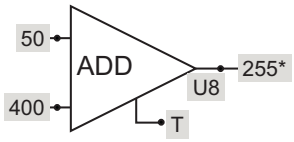
- X1 = A1 + A2
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

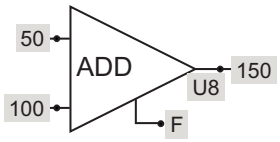
ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT	X2	BOOL

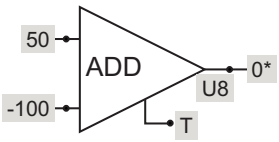
例





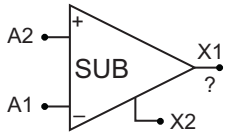
*255—Max value for the U8 data type





*0—Min value for the U8 data type

Subtract Capped（上限付き減算）



用途

- 2つの整数信号を減算します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

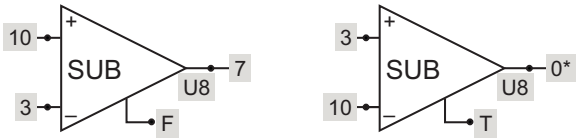
機能

- X1 = A2 – A1
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

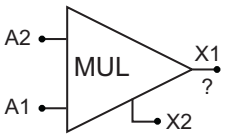
ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT	X2	BOOL

例



*0—Min value for the U8 data type

Multiply Capped（上限付き乗算）



用途

- 2つの整数信号を乗算します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

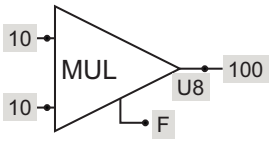
機能

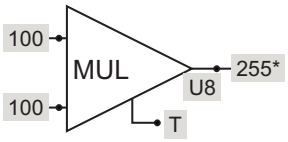
- X1 = A2 ∗ A1
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

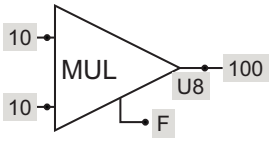
ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT	X2	BOOL

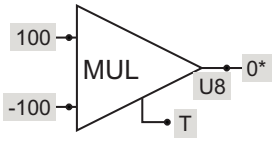
例—Multiply Capped





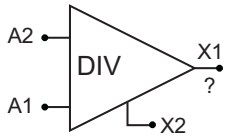
*255—Max value for the U8 data type





*0—Min value for the U8 data type

Divide Capped (上限付き除算)



用途

- 2つの整数信号を除算します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

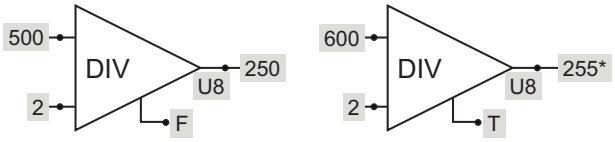
機能

- X1 = A2 ÷ A1
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

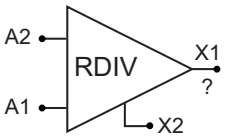
ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT	X2	BOOL

例



*255—Max value for the U8 data type

Rounded Divide Capped（上限付き丸め除算）

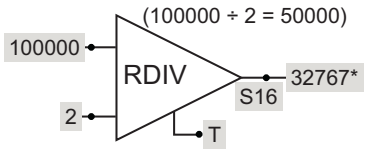


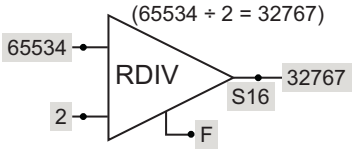
- 2つの整数信号を除算し、丸めた計算結果を出力します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。
- 機能
  - X1 = A2 ÷ A1 の出力を丸めます
    - 剰余が 0.5 以上の場合には切り上げ
    - 剰余が 0.5 未満の場合には切り下げ
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

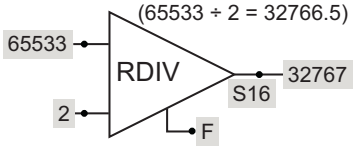
ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT	X2	BOOL

例

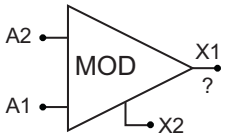




*32767 = Max value for S16 data type



Modulo Capped（上限付き剰余）



用途

- 2つの整数信号の除算から生じるモジュロ（剰余）を出力します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

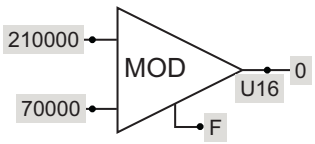
機能

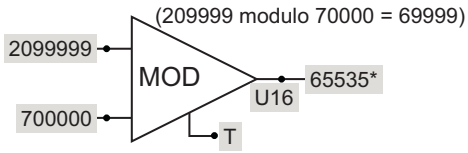
- $X1 = A2 - \left(\left[\frac{A2}{A1} \right] \times A1 \right)$
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

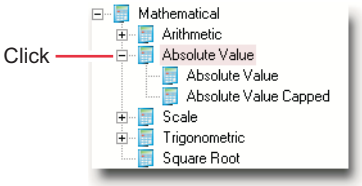
例



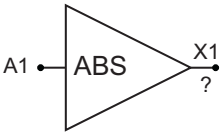


*65535 = Max value for the U16 data type

Absolute Value（絶対値）



Absolute Value（絶対値）



用途

- 整数信号の大きさを出力します。

機能

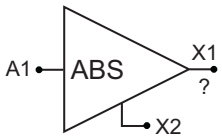
- X1 = A1 の絶対値

有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT

例

Absolute Value Capped（上限付き絶対値）



用途

- 整数信号の大きさを出力します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

機能

- X1 = A1 の絶対値
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

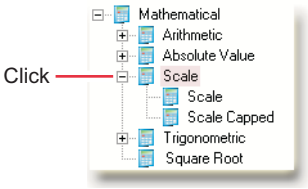
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
		X2	BOOL

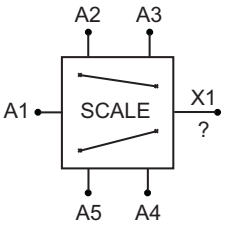
例

*255—Max value for the U8 data type

Scale（スケーリング）



Scale（スケーリング）



用途

- 整数信号を新たな範囲にスケーリングします。
- スケーリングをセンサ信号に適用します。

機能

- $$X1 = \left(\left(\frac{A1 - A5}{A2 - A5} \right) \times \left(A3 - A4 \right) \right) + A4$$

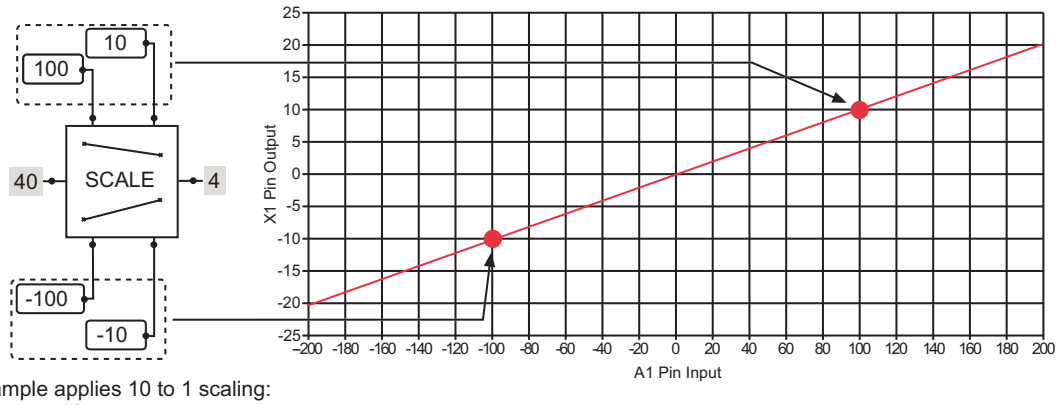
A2-A5 値によって、A1 入力信号に適用される入出力比とオフセットが設定されます。これらによって、結果として生じる X1 出力値が制限されることはありません。

有効接続

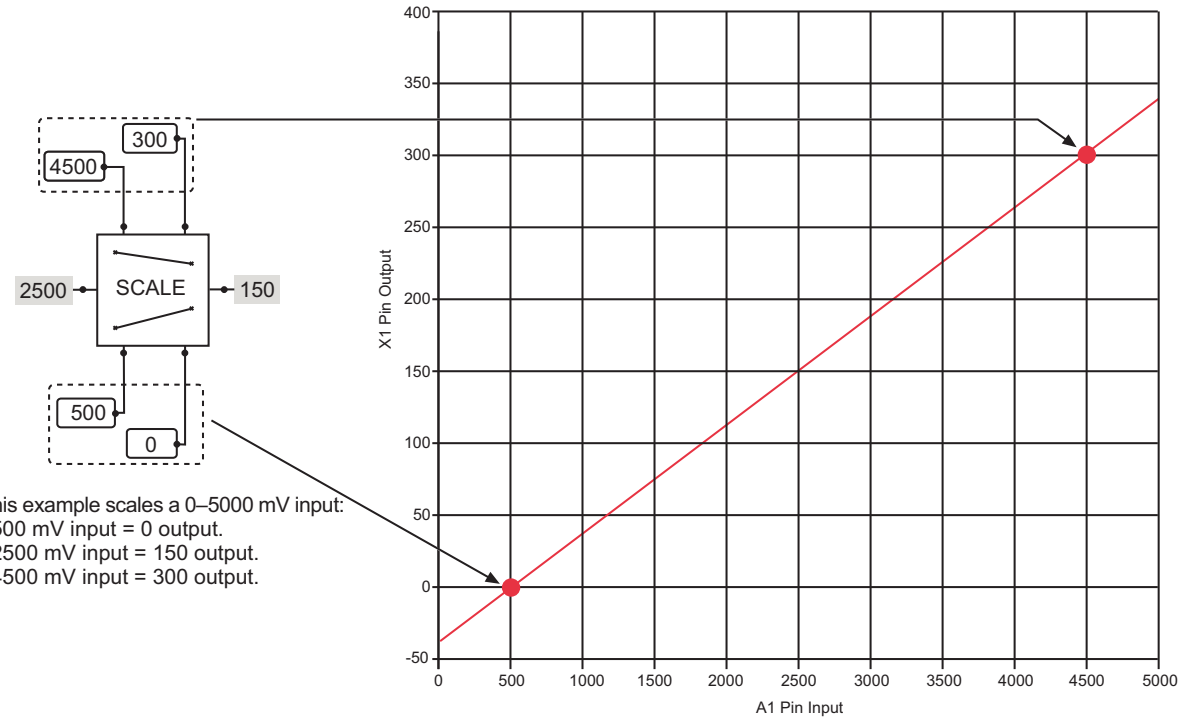
ピン	データ型	ピン	データ型
A1–A5	INT	X1	INT

例

以下の2つの図とプロットは本コンポーネントに適用される様々な入力値が様々な入出力比をもたらす様子を示しています。この例は、両方の **Scale** コンポーネントに適用されます。

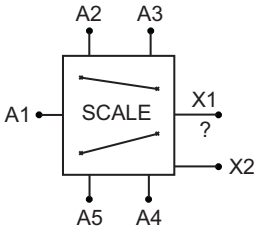


This example applies 10 to 1 scaling:
- -100 input = -10 output.
- 40 input = 4 output.
- 100 input = 10 output.



This example scales a 0–5000 mV input:
- 500 mV input = 0 output.
- 2500 mV input = 150 output.
- 4500 mV input = 300 output.

Scale Capped（上限付きスケーリング）



用途

- 整数信号を新たな範囲にスケーリングします。
- スケーリングをセンサ信号に適用します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

機能

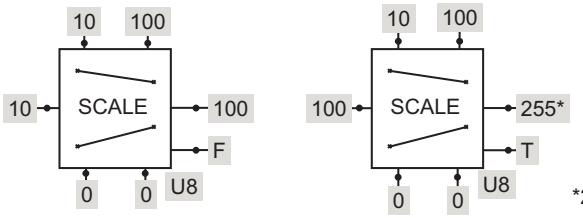
- $$X1 = \left(\left(\frac{A1 - A5}{A2 - A5} \right) \times \left(A3 - A4 \right) \right) + A4$$
- X1 がアンダーフローもオーバーフローもしない場合には、X2 = F となります
- X1 がアンダーフローもしくはオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
- X2 は各プログラムループの開始時に F にリセットします

A2-A5 値によって、A1 入力信号に適用される入出力比とオフセットが設定されます。これらによって、結果として生じる X1 出力値が制限されることはありません。

有効接続

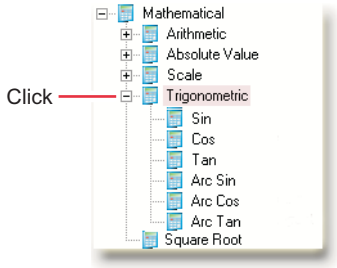
ピン	データ型	ピン	データ型
A1–A5	INT	X1	INT
		X2	BOOL

例

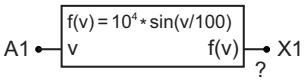


*255—Max value for the U8 data type

Trigonometric（三角関数）



Sin（正弦）



用途

- 整数信号の正弦関数を出力します。

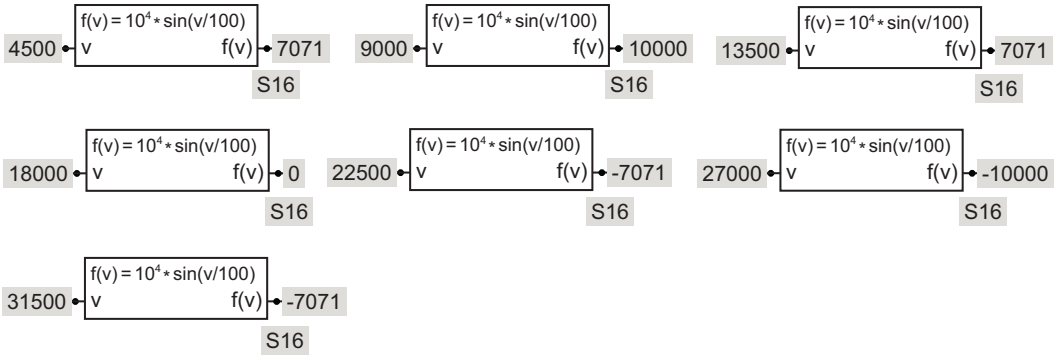
機能

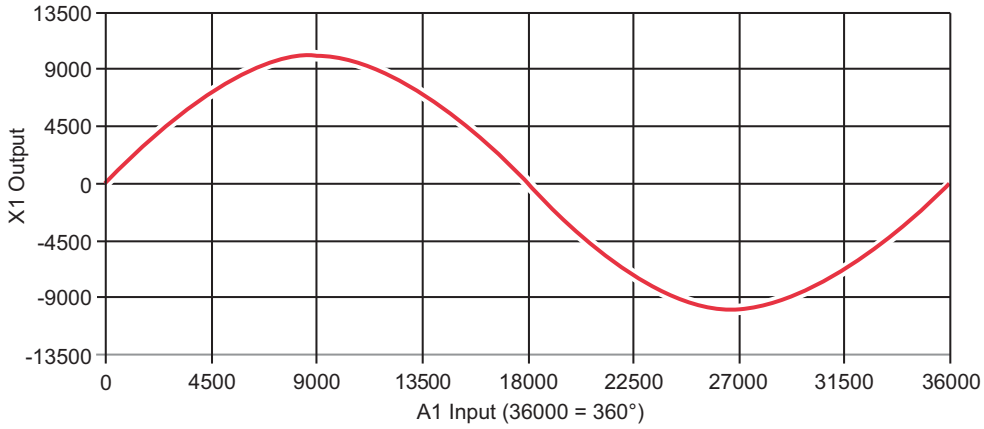
- X1 = 10^4 x sin (A1/100)

有効接続

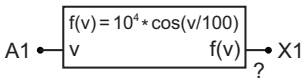
ピン	データ型	ピン	データ型
A1	UINT	X1	INT

例




13500
9000
4500
0
-4500
-9000
-13500
X1 Output
0
4500
9000
13500
18000
22500
27000
31500
36000
A1 Input (36000 = 360°)

Cos（余弦）



用途

- 整数信号の余弦関数を出力します。

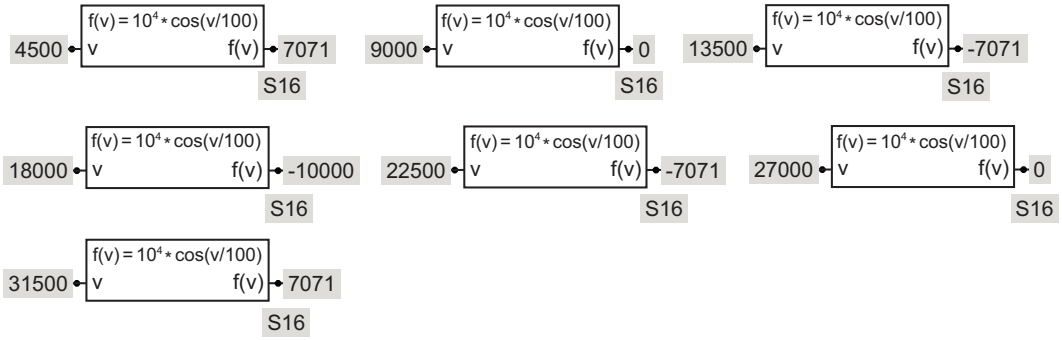
機能

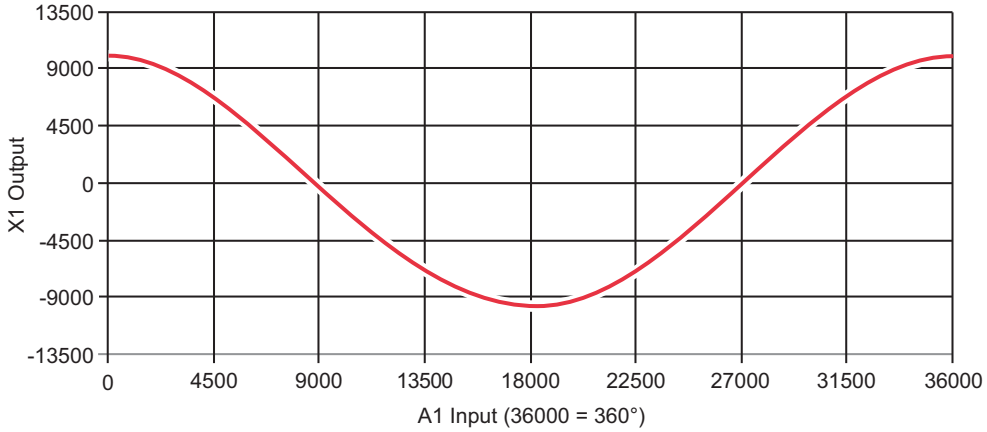
- X1 = 10^4 x cos (A1/100)

有効接続

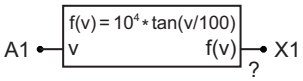
ピン	データ型	ピン	データ型
A1	UINT	X1	INT

例





Tan（正接）



用途

- 整数信号の正接関数を出力します。

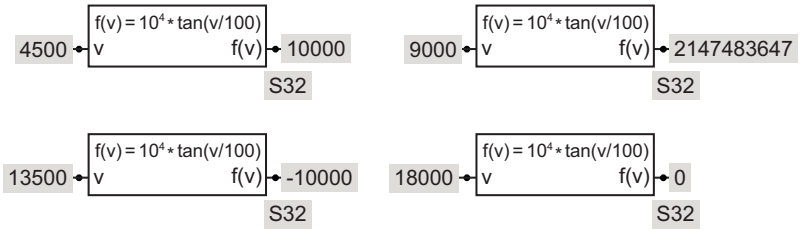
機能

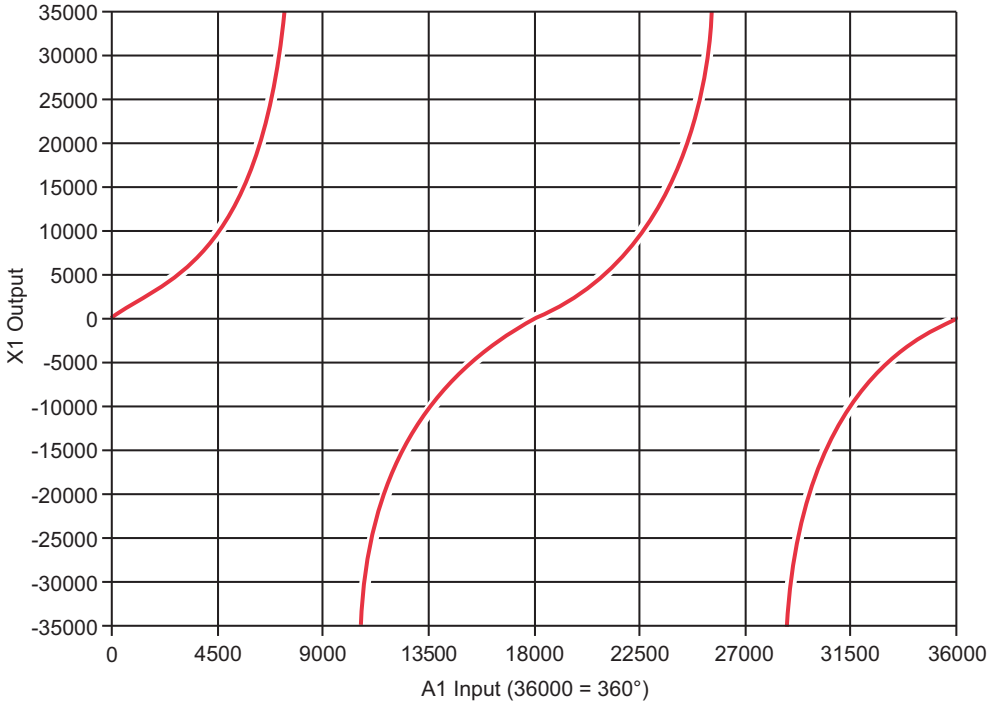
- X1 = 10^4 x tan(A1/100)

有効接続

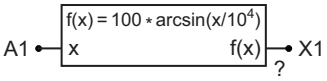
ピン	データ型	ピン	データ型
A1	INT	X1	INT

例





Arc Sin（逆正弦関数）



用途

- 整数信号の逆正弦関数を出力します。

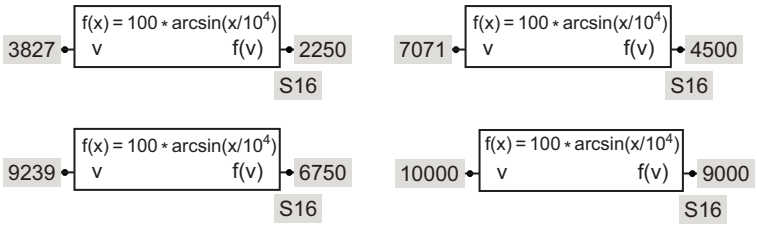
機能

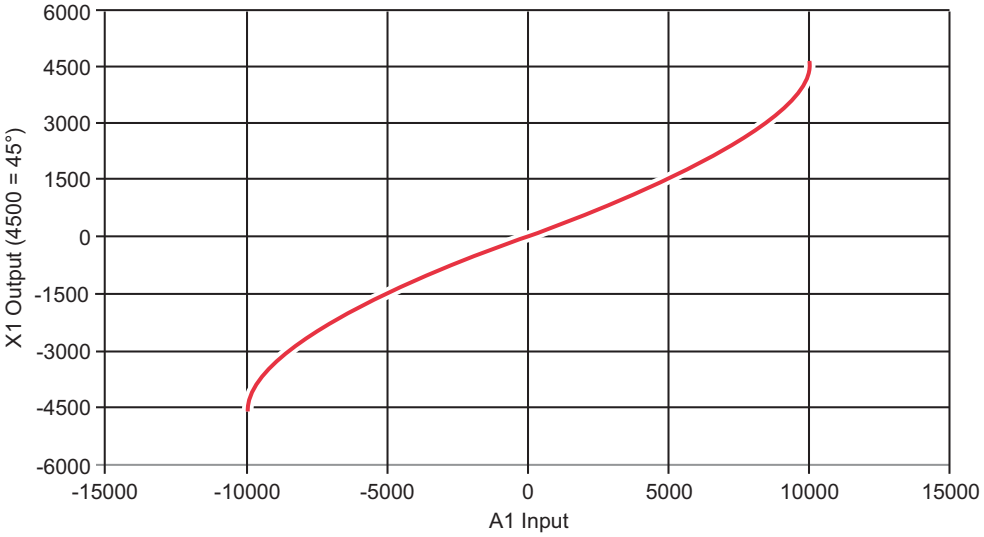
- X1 = 100 x arcsin (A1/10^4)

有効接続

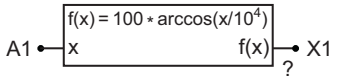
ピン	データ型	ピン	データ型
A1	INT	X1	INT

例





Arc Cos（逆余弦関数）



用途

- 整数信号の逆余弦関数を出力します。

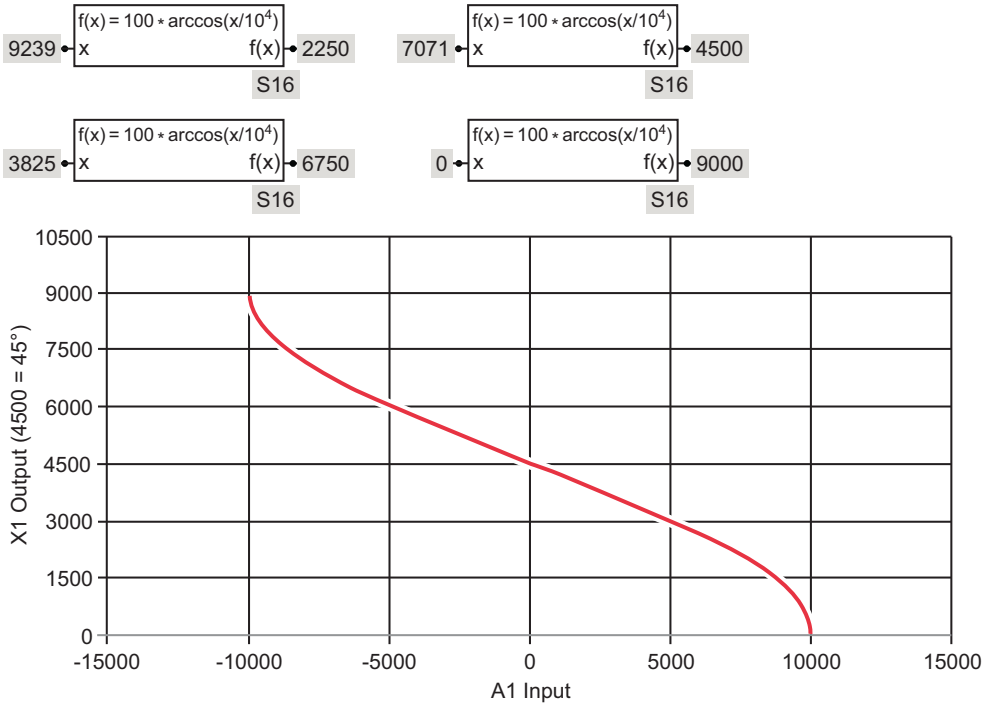
機能

- X1 = 100 x arccos (A1/10^4)

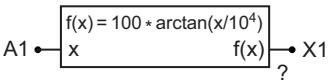
有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	INT

例



Arc Tan（逆正接関数）



用途

- 整数信号の逆正接関数を出力します。

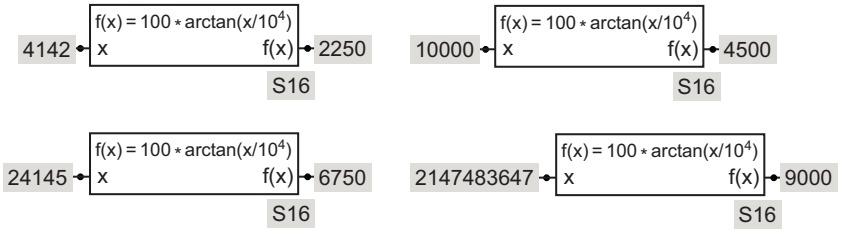
機能

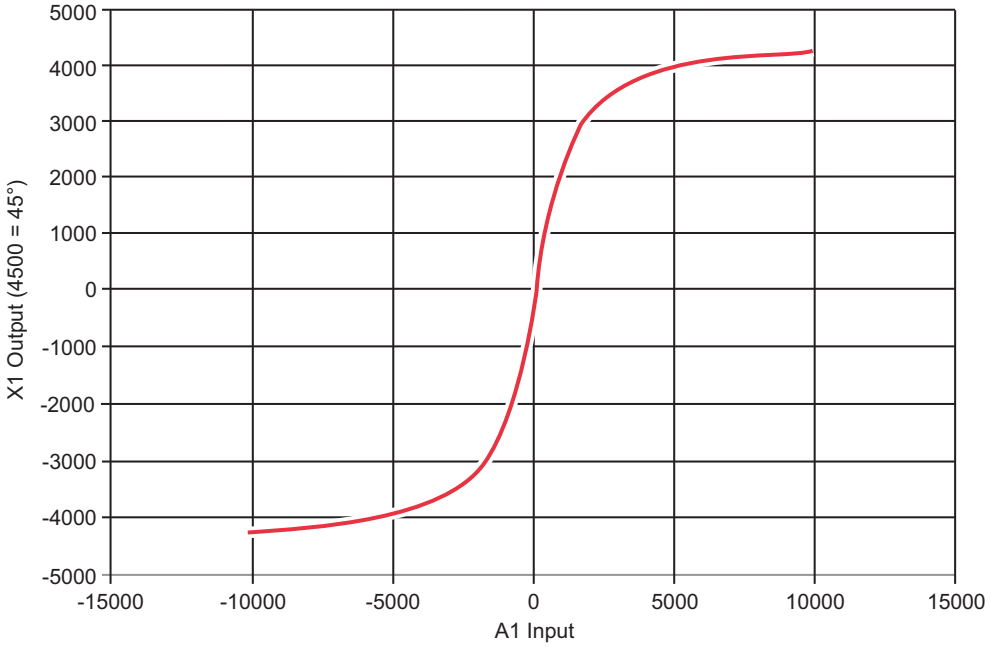
- X1 = 100 x arctan (A1/10^4)

有効接続

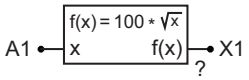
ピン	データ型	ピン	データ型
A1	UINT	X1	INT

例





Square Root（平方根）



用途

- 整数信号の平方根を出力します。

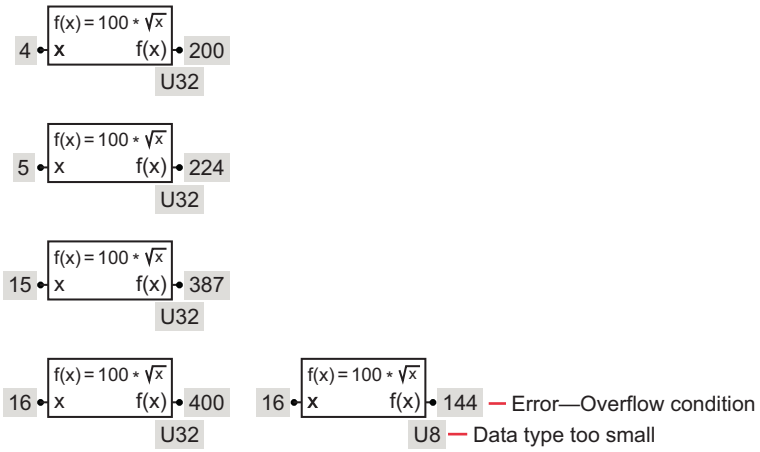
機能

- X1 = 100 x $\sqrt{A1}$
- 精度：±4

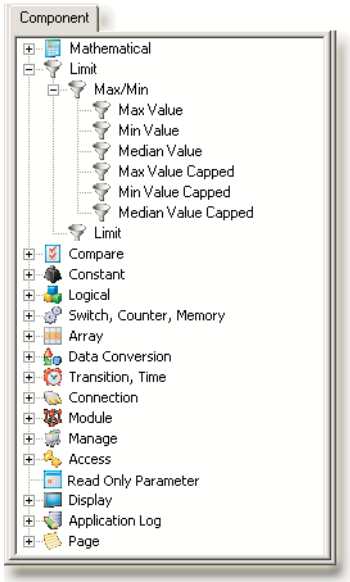
有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	UINT

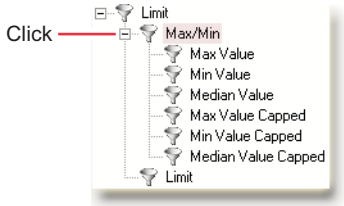
例



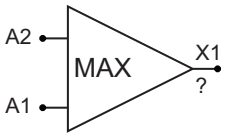
Limit（リミット）


Component
Mathematical
Limit
Max/Min
Max Value
Min Value
Median Value
Max Value Capped
Min Value Capped
Median Value Capped
Limit
Compare
Constant
Logical
Switch, Counter, Memory
Array
Data Conversion
Transition, Time
Connection
Module
Manage
Access
Read Only Parameter
Display
Application Log
Page

Max/Min（最大/最小）



Max Value（最大値）



用途

- 2 つの整数信号の大きい方の値を出力します。

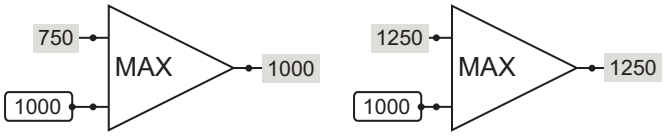
機能

- A1 ≥ A2 であれば X1 = A1
- A2 > A1 であれば X1 = A2

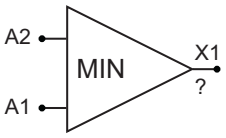
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例



Min Value（最小値）



用途

- 2 つの整数信号の小さい方の値を出力します。

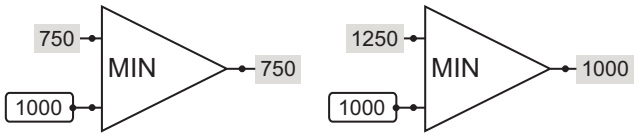
機能

- A1 ≤ A2 であれば X1 = A1
- A2 < A1 であれば X1 = A2

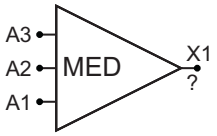
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例



Median Value（中央値）



用途

- 3 つの入力値の中央値を出力します。
- 出力の上限と下限を設定します。

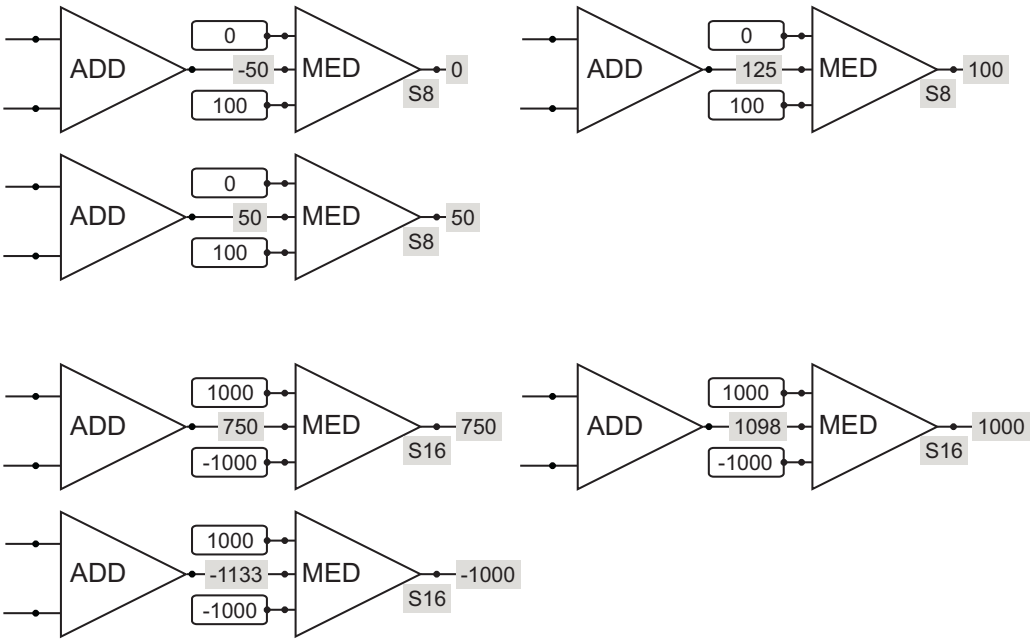
機能

- 以下の場合には X1 = A1
  - A2 ≤ A1 ≤ A3
  - A3 ≤ A1 ≤ A2
- 以下の場合には X1 = A2
  - A1 ≤ A2 ≤ A3
  - A3 ≤ A2 ≤ A1
- 以下の場合には X1 = A3
  - A1 ≤ A3 ≤ A2
  - A2 ≤ A3 ≤ A1

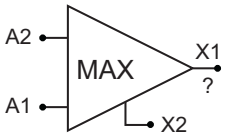
有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	UINT
A2	UINT		
A3	UINT		

例


ADD
0
-50
100
MED
0
S8
ADD
0
125
100
MED
100
S8
ADD
0
50
100
MED
50
S8
ADD
1000
750
-1000
MED
750
S16
ADD
1000
1098
-1000
MED
1000
S16
ADD
1000
-1133
-1000
MED
-1000
S16

Max Value Capped（上限付き最大値）



用途

- 2 つの整数信号の大きい方の値を出力します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

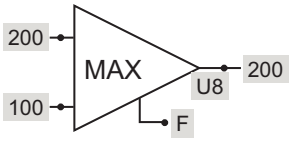
機能

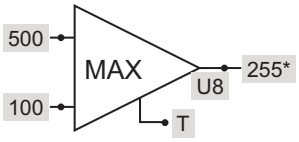
- A1 ≥ A2 であれば X1 = A1
- A2 > A1 であれば X1 = A2
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT	X2	BOOL

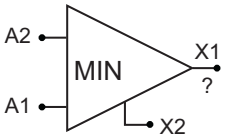
例





*255—Max value for the U8 data type

Min Value Capped（上限付き最小値）



用途

- 2 つの整数信号の小さい方の値を出力します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

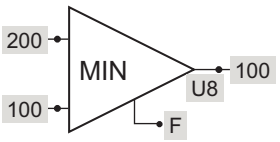
機能

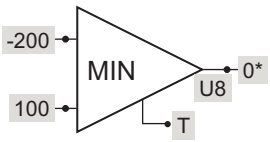
- A1 ≤ A2 であれば X1 = A1
- A2 < A1 であれば X1 = A2
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT	X2	BOOL

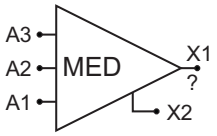
例





*0—Min value for the U8 data type

Median Value Capped（上限付き中央値）



用途

- 3 つの入力値の中央値を出力します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。
- 出力の上限と下限を設定します。

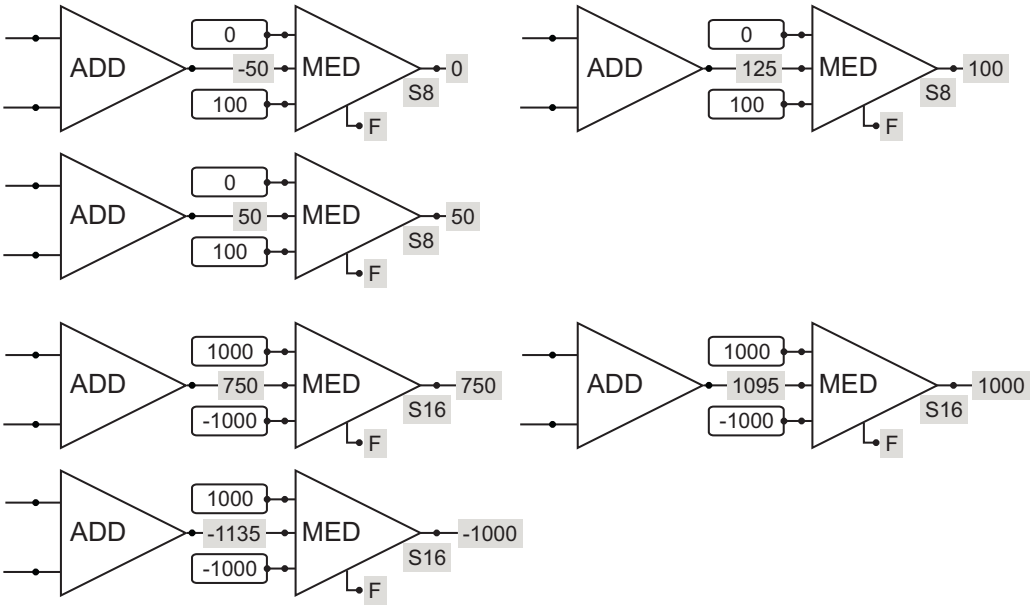
機能

- 以下の場合には X1 = A1
  - A2 ≤ A1 ≤ A3
  - A3 ≤ A1 ≤ A2
- 以下の場合には X1 = A2
  - A1 ≤ A2 ≤ A3
  - A3 ≤ A2 ≤ A1
- 以下の場合には X1 = A3
  - A1 ≤ A3 ≤ A2
  - A2 ≤ A3 ≤ A1
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

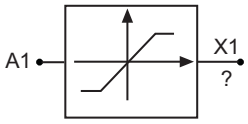
有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	UINT
A2	UINT	X2	BOOL
A3	UINT		

例



Limit（制限）



用途:

- X1 出力を X1 に選択されているデータ型の最大値と最小値に制限します。

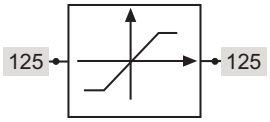
機能

- X1 データ型の範囲内である場合には X1 = A1
- A1 が X1 データ型の最大値を上回る場合には X1 =X1 データ型の最大値
- A1 が X1 データ型の最小値を下回る場合には X1 =X1 データ型の最小値

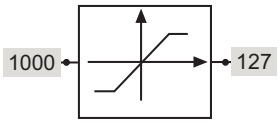
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT

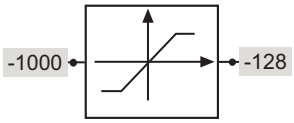
例



(S16 in/S8 out)

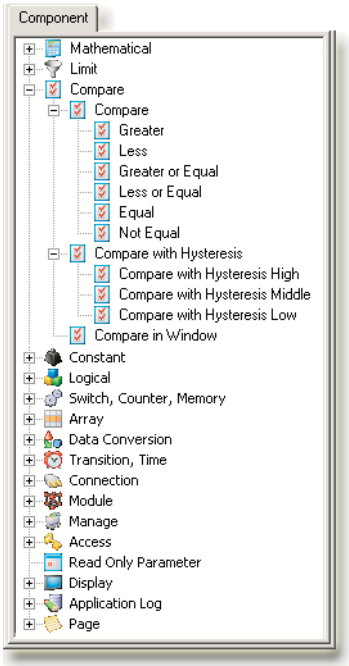


(S16 in/S8 out)

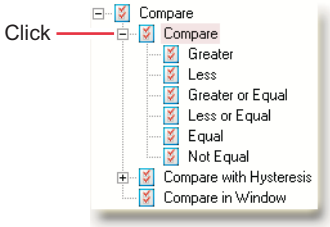


(S16 in/S8 out)

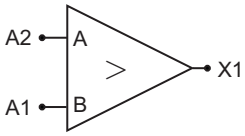
Compare（比較）


Component
Mathematical
Limit
Compare
Compare
Greater
Less
Greater or Equal
Less or Equal
Equal
Not Equal
Compare with Hysteresis
Compare with Hysteresis High
Compare with Hysteresis Middle
Compare with Hysteresis Low
Compare in Window
Constant
Logical
Switch, Counter, Memory
Array
Data Conversion
Transition, Time
Connection
Module
Manage
Access
Read Only Parameter
Display
Application Log
Page

Compare（比較）



Greater（超過）



用途

- A が B を超過していれば T を出力します。

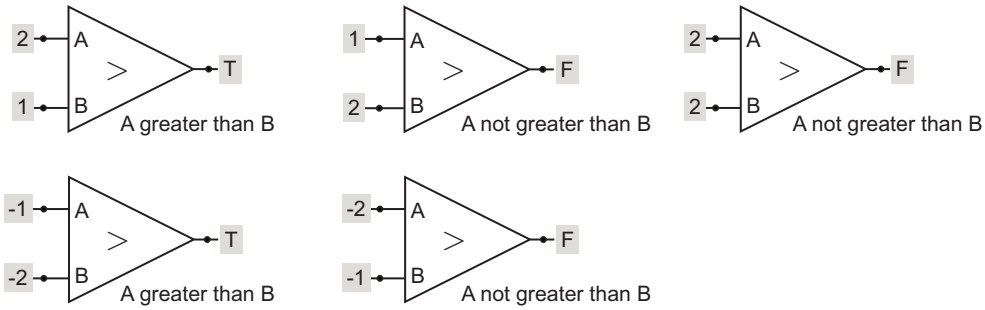
機能

- A2 を A1 と比較して、
  - A2 > A1 であれば X1 = T
  - A2 = A1 であれば X1 = F
  - A2 < A1 であれば X1 = F

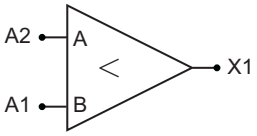
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		

例



Less（未満）



用途

- A が B 未満であれば T を出力します。

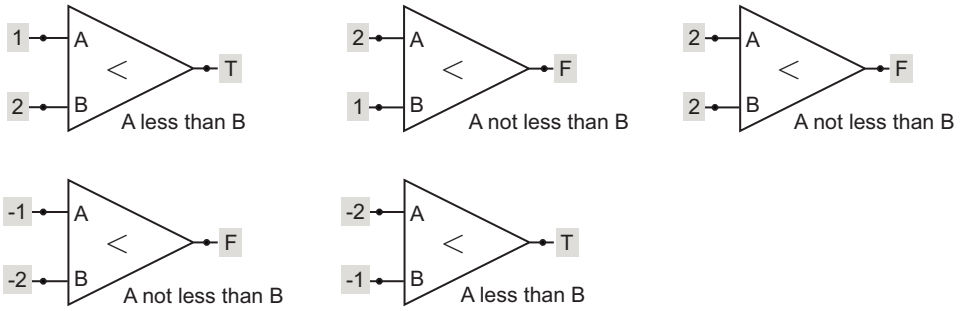
機能

- A2 を A1 と比較して、
  - A2 < A1 であれば X1 = T
  - A2 = A1 であれば X1 = F
  - A2 > A1 であれば X1 = F

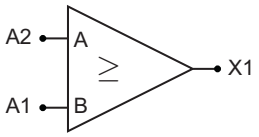
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		

例



Greater or Equal（以上）



用途

- A が B 以上であれば T を出力します。

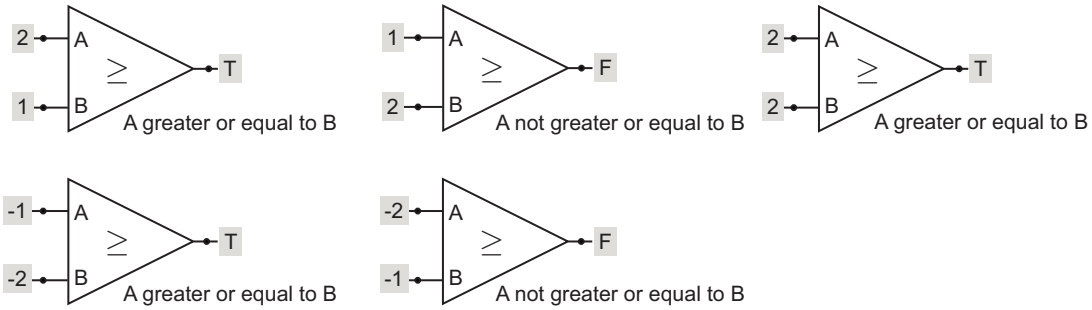
機能

- A2 を A1 と比較して、
  - A2 < A1 であれば X1 = F
  - A2 = A1 であれば X1 = T
  - A2 > A1 であれば X1 = T

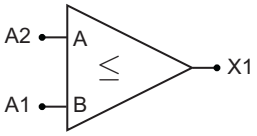
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		

例



Less or Equal（以下）



用途

- A が B 以下であれば T を出力します。

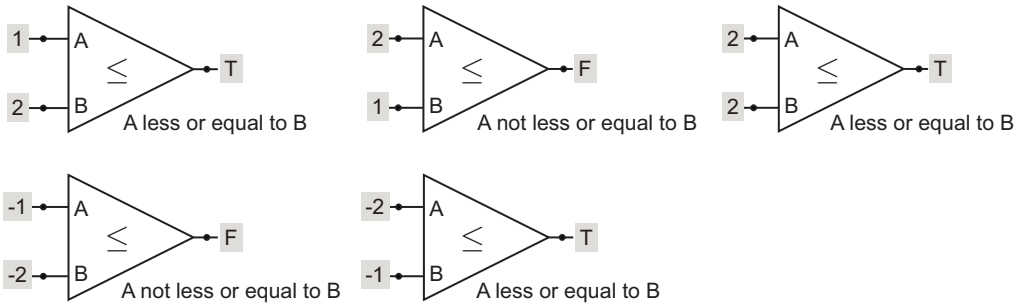
機能

- A2 を A1 と比較して、
  - A2 < A1 であれば X1 = T
  - A2 = A1 であれば X1 = T
  - A2 > A1 であれば X1 = F

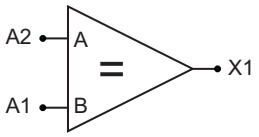
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		

例



Equal（等価）



用途

- A が B と等しければ T を出力します。

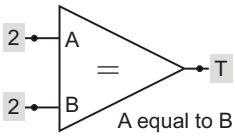
機能

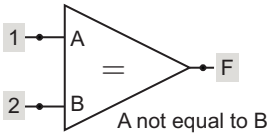
- A2 を A1 と比較して、
  - A2 < A1 であれば X1 = F
  - A2 = A1 であれば X1 = T
  - A2 > A1 であれば X1 = F

有効接続

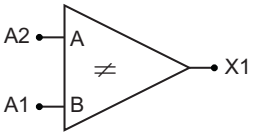
ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		

例





Not Equal（不等）



用途

- A が B と等しくなければ T を出力します。

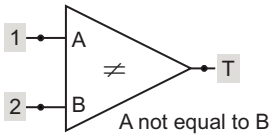
機能

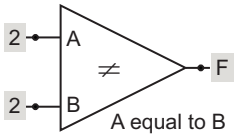
- A2 を A1 と比較して、
  - A2 ≠ A1 であれば X1 = F
  - A2 = A1 であれば X1 = T

有効接続

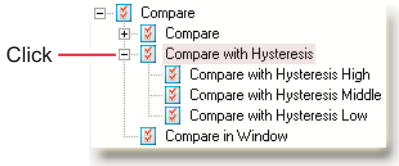
ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		

例

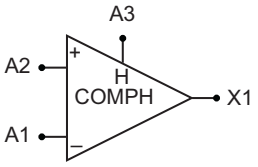




Compare with Hysteresis（ヒステリシス付き比較）



Compare with Hysteresis High（高位ヒステリシス付き比較）



用途

- 2 つの整数入力信号間のヒステリシス比較に基づいてブール値を出力します。

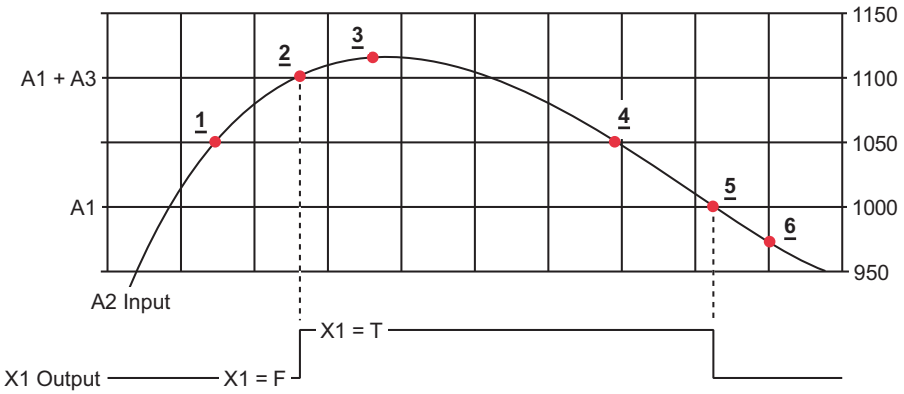
機能

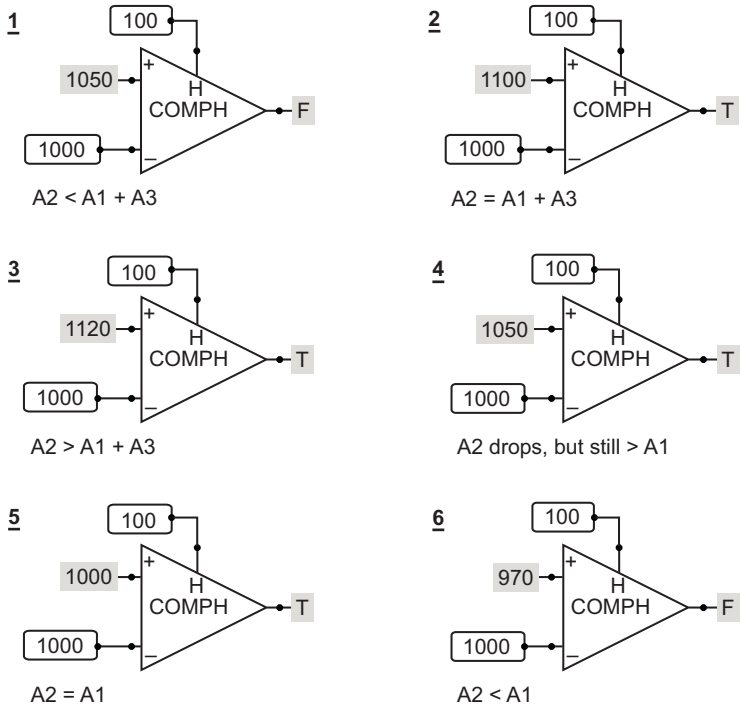
- A2 ≥ (A1 + A3)であれば X1 = T
- A2 < A1 であれば X1 = F
- A1 ≤ A2 < (A1 + A3) であれば X1 = 無変化(No Change)

有効接続

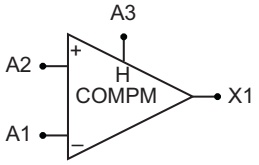
ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		
A3	UINT		

例


1150
1100
1050
1000
950
A1 + A3
A1
1
2
3
4
5
6
A2 Input
X1 = T
X1 Output
X1 = F


1
100
1050
1000
H
COMPH
F
A2 < A1 + A3
2
100
1100
1000
H
COMPH
T
A2 = A1 + A3
3
100
1120
1000
H
COMPH
T
A2 > A1 + A3
4
100
1050
1000
H
COMPH
T
A2 drops, but still > A1
5
100
1000
1000
H
COMPH
T
A2 = A1
6
100
970
1000
H
COMPH
F
A2 < A1

Compare with Hysteresis Middle（中位ヒステリシス付き比較）



用途

- 2 つの整数入力信号間のヒステリシス比較に基づいてブール値を出力します。

機能

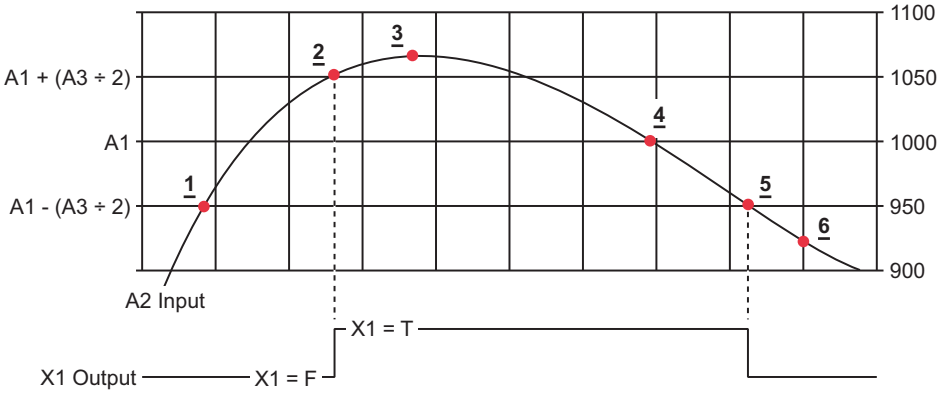
- A2 ≥ A1 + (A3 ÷ 2)であれば X1 = T
- A2 < A1 – (A3 ÷ 2)であれば X1 = F
- A1 – (A3 ÷ 2) ≤ A2 < A1 + (A3 ÷ 2)であれば X1 = 無変化

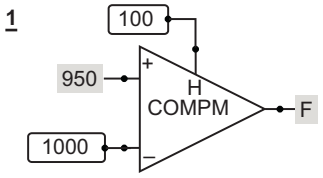
  A3 ÷ 2 の商は次に低い整数に切り下げられます（例えば 7.5 は 7 になります）。

有効接続

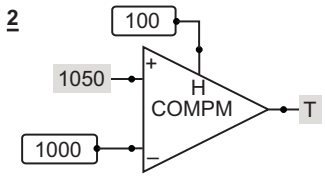
ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		
A3	UINT		

例

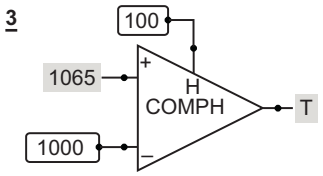




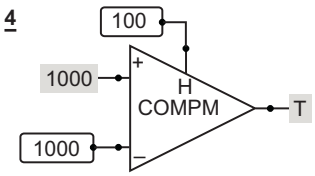
A2 < A1 + (A3 ÷ 2)



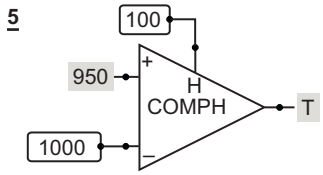
A2 = A1 + (A3 ÷ 2)



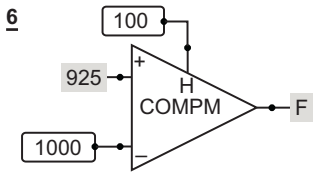
A2 > A1 - (A3 ÷ 2)



A2 drops, but still > A1 - (A3 ÷ 2)

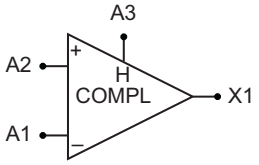


A2 = A1 - (A3 ÷ 2)



A2 < A1 - (A3 ÷ 2)

Compare with Hysteresis Low（低位ヒステリシス付き比較）



用途

- 2 つの整数入力信号間のヒステリシス比較に基づいてブール値を出力します。

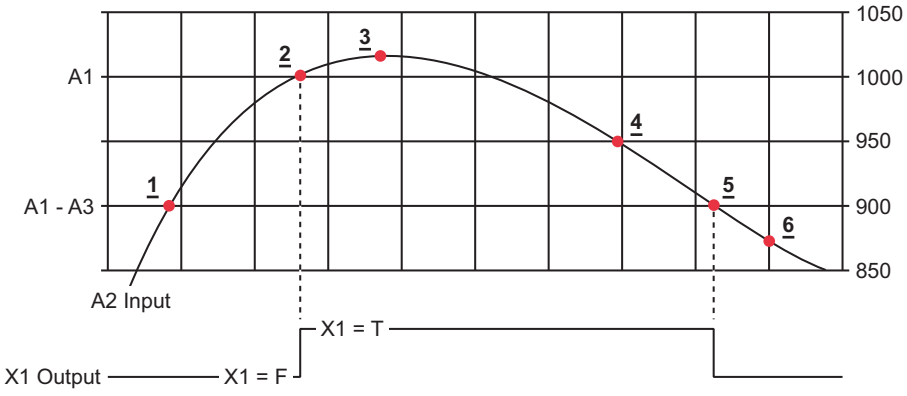
機能

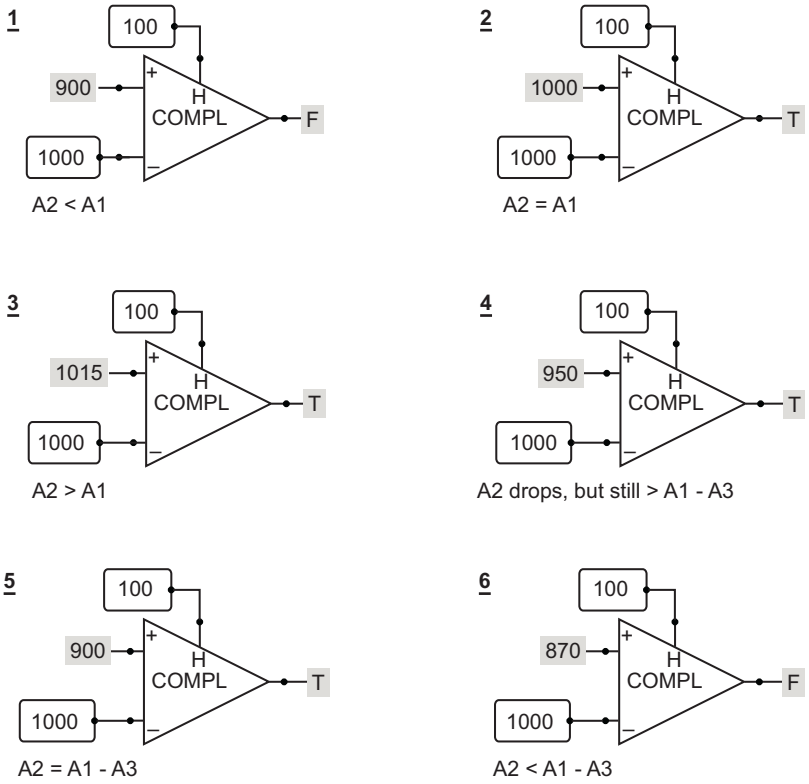
- A2 ≥ A1 であれば X1 = T
- A2 < (A1 − A3) であれば X1 = F
- (A1 – A3) ≤ A2 < A1 であれば X1 = 無変化

有効接続

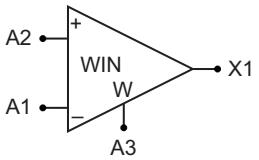
ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		
A3	UINT		

例


1050
1000
950
900
850
A1
A1 - A3
1
2
3
4
5
6
A2 Input
X1 = T
X1 Output
X1 = F


1
100
900
1000
H
COMPL
F
A2 < A1
2
100
1000
1000
H
COMPL
T
A2 = A1
3
100
1015
1000
H
COMPL
T
A2 > A1
4
100
950
1000
H
COMPL
T
A2 drops, but still > A1 - A3
5
100
900
1000
H
COMPL
T
A2 = A1 - A3
6
100
870
1000
H
COMPL
F
A2 < A1 - A3

Compare in Window（ウィンドウ内比較）



用途

- 2 つの信号間の絶対差が指定範囲内にあるかチェックし、この比較に基づいてブール値を出力します。

機能

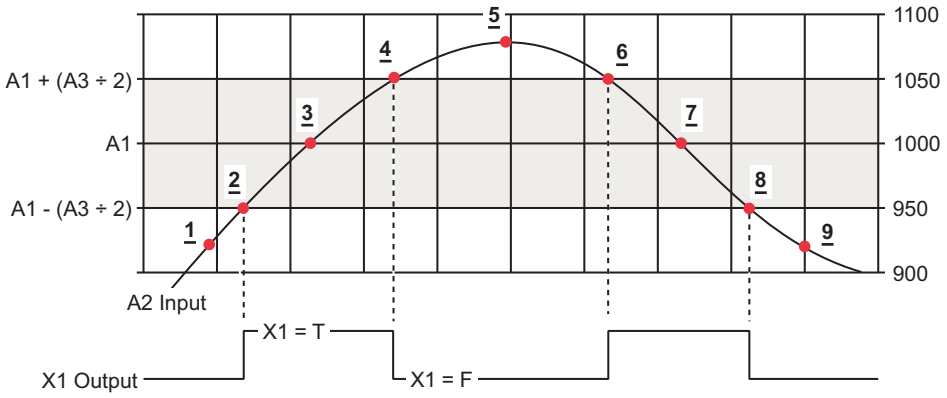
- A2 ≥ A1 – (A3 ÷ 2) および A2 ≤ A1 + (A3 ÷ 2)であれば X1 = T
- A2 > A1 + (A3 ÷ 2) であれば X1 = F
- A2 < A1 – (A3 ÷ 2) であれば X1 = F

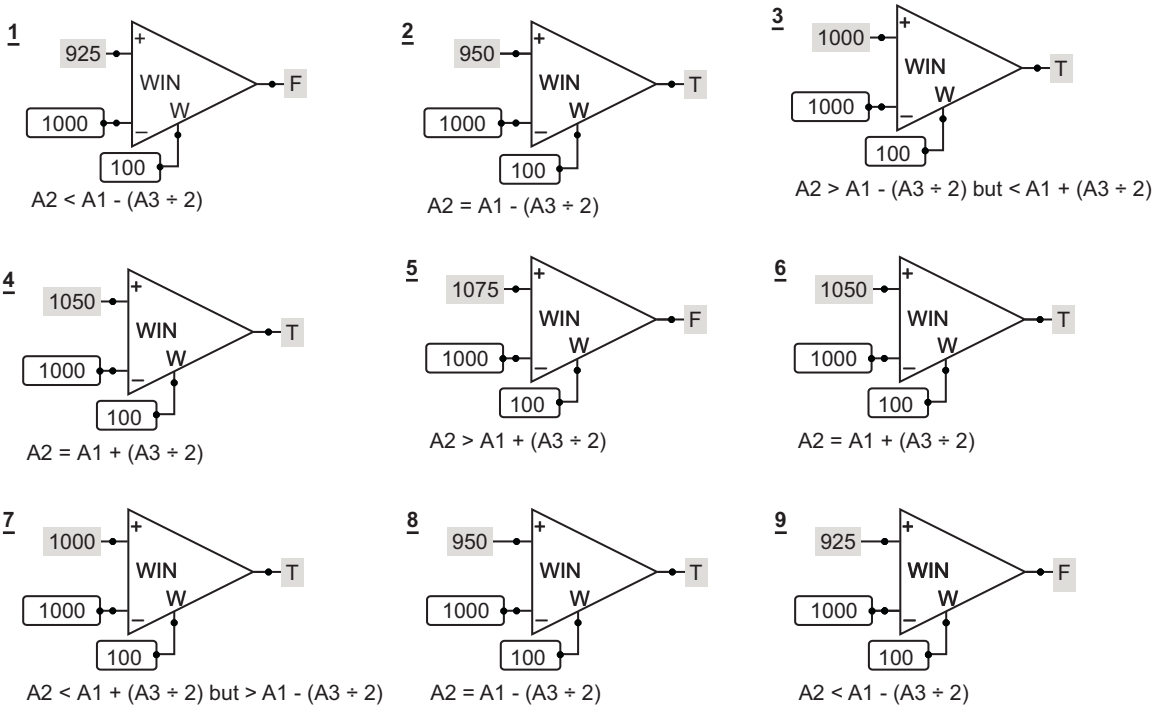
A3 ÷ 2 の商は次に低い整数に切り下げられます（例えば 7.5 は 7 になります）。

有効接続

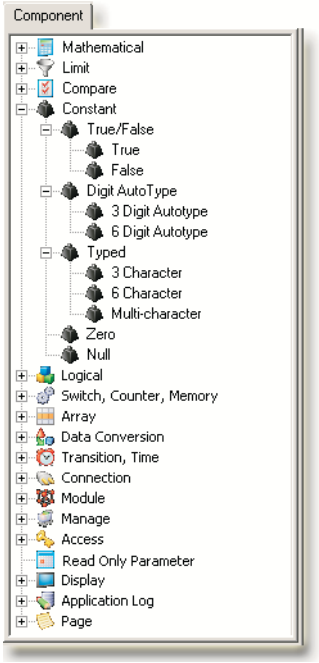
ピン	データ型	ピン	データ型
A1	INT	X1	BOOL
A2	INT		
A3	UINT		

例

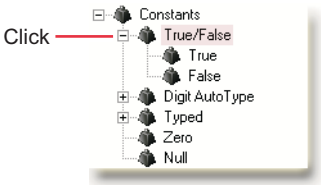

1100
1050
1000
950
900
A1 + (A3 ÷ 2)
A1
A1 - (A3 ÷ 2)
1
2
3
4
5
6
7
8
9
A2 Input
X1 = T
X1 = F
X1 Output


1
925
1000
100
WIN
W
F
A2 < A1 - (A3 ÷ 2)
2
950
1000
100
WIN
W
T
A2 = A1 - (A3 ÷ 2)
3
1000
1000
100
WIN
W
T
A2 > A1 - (A3 ÷ 2) but < A1 + (A3 ÷ 2)
4
1050
1000
100
WIN
W
T
A2 = A1 + (A3 ÷ 2)
5
1075
1000
100
WIN
W
F
A2 > A1 + (A3 ÷ 2)
6
1050
1000
100
WIN
W
T
A2 = A1 + (A3 ÷ 2)
7
1000
1000
100
WIN
W
T
A2 < A1 + (A3 ÷ 2) but > A1 - (A3 ÷ 2)
8
950
1000
100
WIN
W
T
A2 = A1 - (A3 ÷ 2)
9
925
1000
100
WIN
W
F
A2 < A1 - (A3 ÷ 2)

Constant（定数）


Component
Mathematical
Limit
Compare
Constant
True/False
True
False
Digit AutoType
3 Digit Autotype
6 Digit Autotype
Typed
3 Character
6 Character
Multi-character
Zero
Null
Logical
Switch, Counter, Memory
Array
Data Conversion
Transition, Time
Connection
Module
Manage
Access
Read Only Parameter
Display
Application Log
Page

True/False（真/偽）



True（真）

用途

- ブール定数の True 値を出力します。

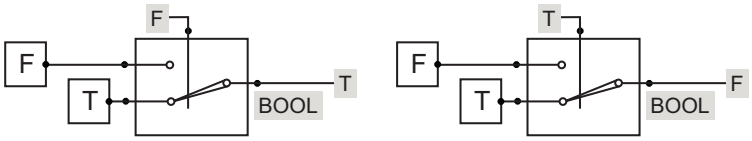
機能

- X1 = T

有効接続

ピン	データ型	ピン	データ型
—	—	X1	BOOL

例



False（偽）

F X1

用途

- ブール定数の False 値を出力します。

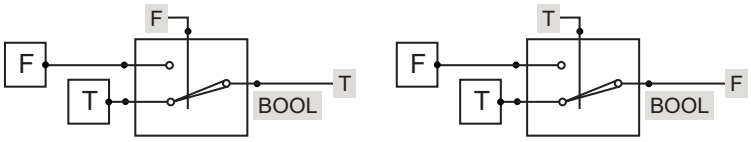
機能

- X1 = F

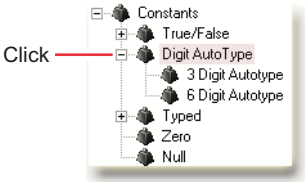
有効接続

ピン	データ型	ピン	データ型
—	—	X1	BOOL

例



Digit Autotype（桁オートタイプ）



3 Digit Autotype（3 桁オートタイプ）

用途

- 定数値を出力します。
- 3 桁までの値のためにサイズ設定されています。
- 入力された値によって、データ型が設定されます（下表参照）。
- 数字のみを受け付けます。

ブール値を出力する場合は、**False** コンポーネントと **True** コンポーネントを使用します。

本コンポーネントは、3 桁を超える値を受け付けます。本コンポーネントの境界を越える値は、読み取りが困難となることがあります。そのような値の場合は、***6 Digit Autotype***コンポーネント（***169***ページ参照）か***Multi-Character***コンポーネント（***173***ページ参照）を使用します。

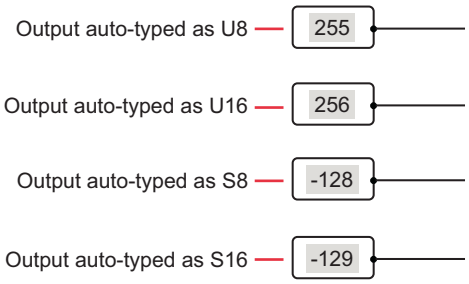
機能

- X1 = ?のオートタイプ値

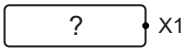
データ型

入力値	データ型
0 ～ +255	U8
−128 ～ −1	S8
+256 ～ +65535	U16
−32768 ～ −129	S16
+65536 ～ +4294967296	U32
−2147483648 ～ −32769	S32

例



6 Digit Autotype（6 桁オートタイプ）



用途

- 定数値を出力します。
- 6 桁までの値のためにサイズ設定されています
- 入力された値によって、自動的にデータ型が設定されます（下表参照）。
- 数字のみを受け付けます。

  ブール値を出力する場合は、**False** コンポーネントと **True** コンポーネントを使用します。

本コンポーネントは、6 桁を超える値や 6 桁未満の値を受け付けます。本コンポーネントの境界を越える値は、読み取りが困難となることがあります。そのような値の場合は、***Multi-Character*** コンポーネント（***173***ページ参照）を使用します。

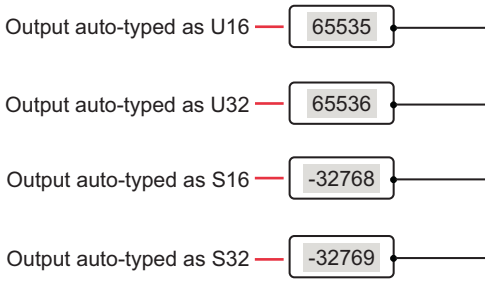
機能

- X1 = ?のオートタイプ値

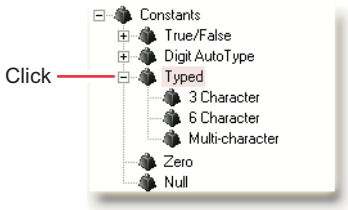
データ型

データ型	データ型
0 ～ +255	U8
−128 ～ −1	S8
+256 ～ +65535	U16
−32768 ～ −129	S16
+65536 ～ +4294967296	U32
−2147483648 ～ −32769	S32

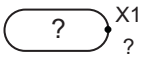
例



Typed（型付き）



3 Character（3 文字）



用途

- 定数値を出力します。
- 値のデータ型を定めなければなりません。
- コンポーネントは 3 桁までの値のためにサイズ設定されています。

本コンポーネントは、3 桁を超える値を受け付けます。本コンポーネントの境界を越える値は、読み取りが困難となることがあります。そのような値の場合は、***6 Character***コンポーネント（***172***ページ参照）か***Multi-Character***コンポーネント（***173***ページ参照）を使用します。

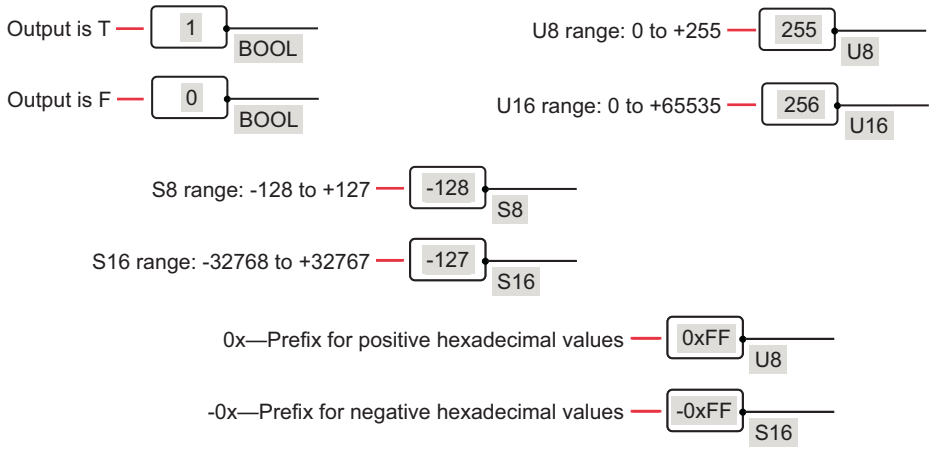
機能

- X1 = ?の値

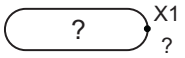
有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

例


Output is T
1
BOOL
Output is F
0
BOOL
U8 range: 0 to +255
255
U8
U16 range: 0 to +65535
256
U16
S8 range: -128 to +127
-128
S8
S16 range: -32768 to +32767
-127
S16
0x—Prefix for positive hexadecimal values
0xFF
U8
-0x—Prefix for negative hexadecimal values
-0xFF
S16

6 Character（6 文字）



用途

- 定数値を出力します。
- 値のデータ型を定めなければなりません。
- コンポーネントは 6 桁までの値用にサイズ設定されています。

本コンポーネントは、6 桁を超える値を受け付けます。本コンポーネントの境界を越える値は、読み取りが困難となることがあります。そのような値の場合は、***Multi-Character***コンポーネント（***173***ページ参照）を使用します。

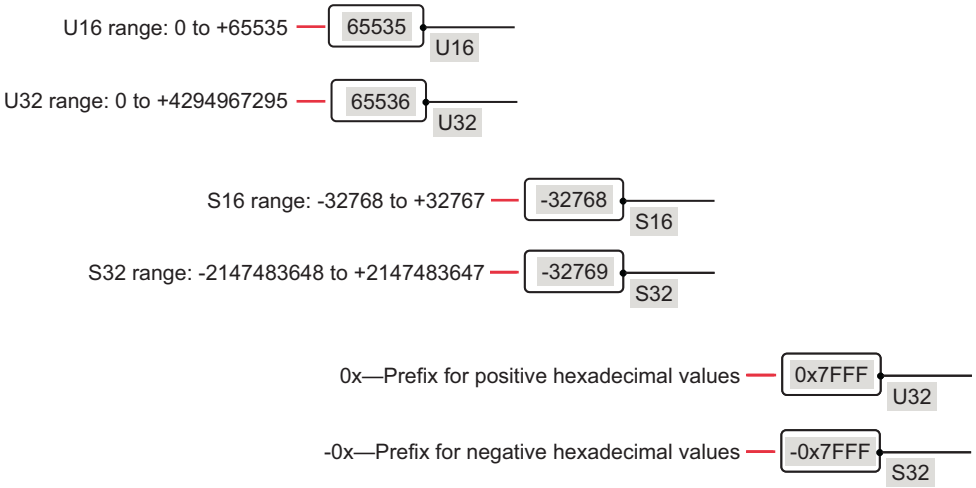
機能

- X1 = ?の値

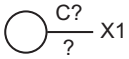
有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

例



Multi-Character（複数文字）



用途

- 定数値を出力します。
- 値のデータ型を定めなければなりません。
- 大きな複数文字定数に使用します。
- 本コンポーネントを用いて配列を入力する方法に関しては、72 ページの「配列データ型」を参照してください。

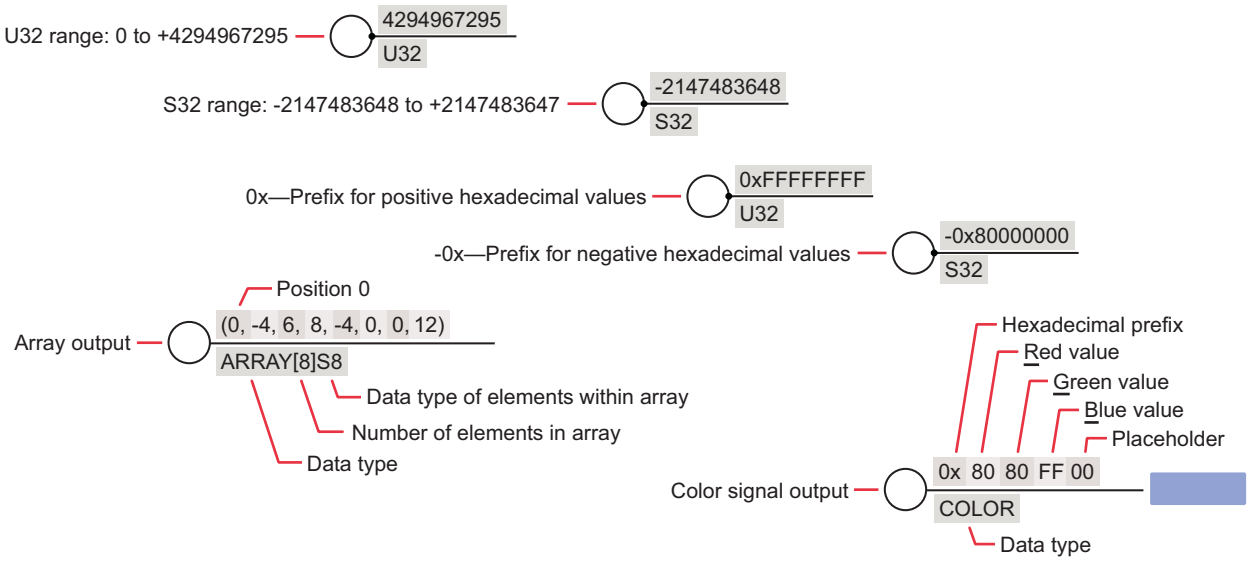
機能

- X1 = C?

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

例



Zero（ゼロ）

用途

- 0 の定数値を定義します。

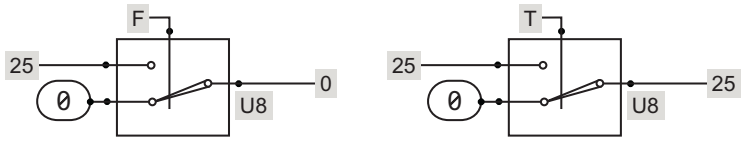
機能

- X1 = 0

有効接続

ピン	データ型	ピン	データ型
—	—	X1	U8

例



Null（ヌル）

用途

- 接続が入力を必要としていないことを示します。
- 使用されていない入力に NULL 値を与えることができます。

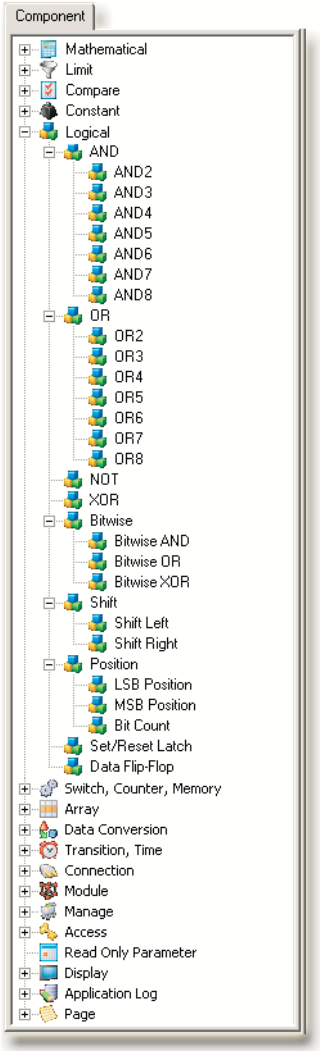
機能

- X1 = NULL

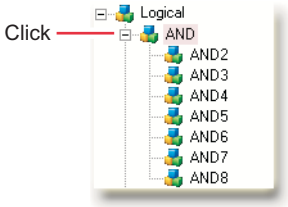
有効接続

ピン	データ型	ピン	データ型
—	—	X1	NULL

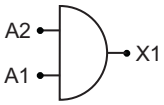
Logical（論理）


Component
Mathematical
Limit
Compare
Constant
Logical
AND
AND2
AND3
AND4
AND5
AND6
AND7
AND8
OR
OR2
OR3
OR4
OR5
OR6
OR7
OR8
NOT
XOR
Bitwise
Bitwise AND
Bitwise OR
Bitwise XOR
Shift
Shift Left
Shift Right
Position
LSB Position
MSB Position
Bit Count
Set/Reset Latch
Data Flip-Flop
Switch, Counter, Memory
Array
Data Conversion
Transition, Time
Connection
Module
Manage
Access
Read Only Parameter
Display
Application Log
Page

AND2–AND8（論理積 2～論理積 8）



入力数の違いを除いて、AND2～AND8 コンポーネントは同様に機能します。ここでは AND2 コンポーネントのみについて説明します。



用途

- 2～8 個のブール信号に対する AND 関数。

機能

- X1 = A1～A8 に AND 関数を作用させたもの

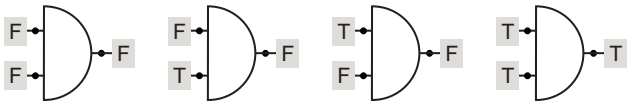
真理値表

A1 入力	A2 入力	X1 出力
F	F	F
T	F	F
F	T	F
T	T	T

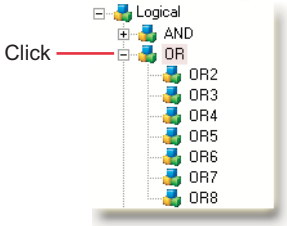
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	BOOL		

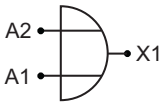
例: AND



OR2–OR8（論理和 2～論理和 8）



入力数の違いを除いて、OR2～OR8 コンポーネントは同様に機能します。ここでは OR2 コンポーネントのみについて説明します。



用途

- 2～8 個のブール信号に対する OR 関数。

機能

- X1 = A1～A8 に OR 関数を作用させたもの

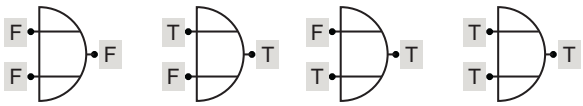
真理値表

A1 入力	A2 入力	X1 出力
F	F	F
T	F	T
F	T	T
T	T	T

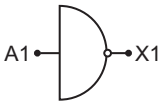
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	BOOL		

例: OR



NOT（否定）



用途

- ブール信号に対する NOT 関数。

機能

- X1 = A1 に NOT 関数を作用させたもの

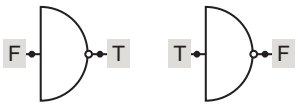
真理値表

A1 入力	X1 出力
F	T
T	F

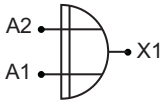
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL

例: NOT



XOR（排他的論理和）



用途

- 2 つのブール信号に対する排他的 OR 関数。

機能

- X1 = A1 と A2 に XOR 関数を作用させたもの

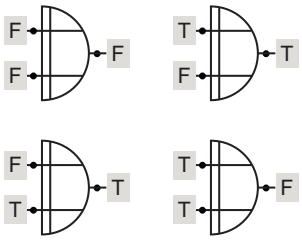
真理値表

A1 入力	A2 入力	X1 出力
F	F	F
T	F	T
F	T	T
T	T	F

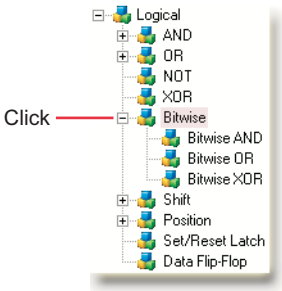
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	BOOL		

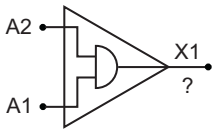
例: XOR



Bitwise（ビット単位）



Bitwise AND（ビット単位の論理積）



用途

- 2 つの整数信号のビット状態のマスクを出力します。

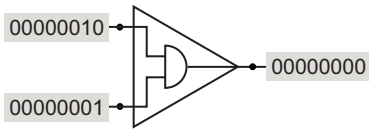
機能

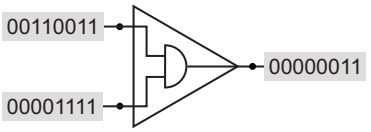
- X1 = A1 と A2 のビット単位 AND 出力

有効接続

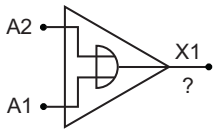
ピン	データ型	ピン	データ型
A1	UINT	X1	UINT
A2	UINT		

例: Bitwise AND





Bitwise OR（ビット単位の論理和）



用途

- 2 つの整数信号のビット状態のマージを出力します。

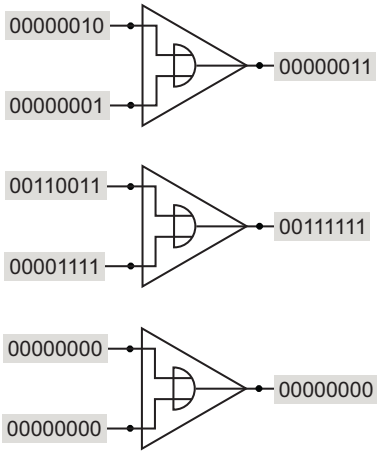
機能

- X1 = A1 と A2 のビット単位 OR 出力

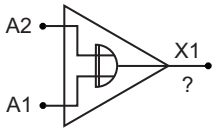
有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	UINT
A2	UINT		

例: Bitwise OR



Bitwise XOR（ビット単位の排他的論理和）



用途

- 2 つの整数信号のビット状態の比較を出力します。

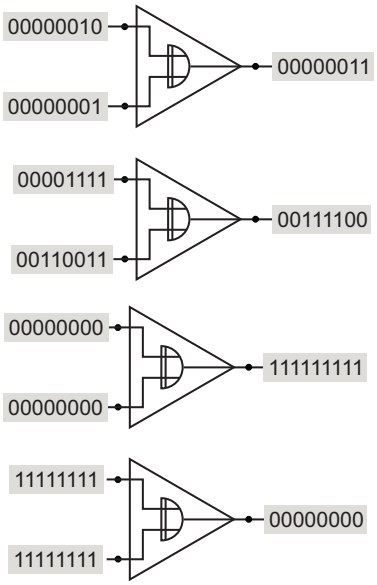
機能

- X1 = A1 と A2 のビット単位 XOR 出力

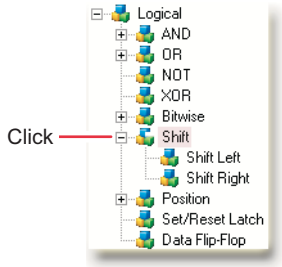
有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	UINT
A2	UINT		

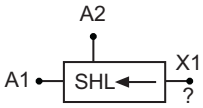
例: Bitwise XOR



Shift（シフト）



Shift Left（左シフト）



用途

- 整数信号のビットデータをシフトします。

機能

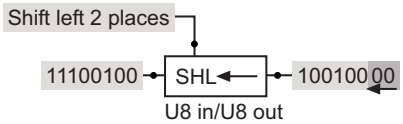
- X1 = A2 の値が示す位置数だけ左にシフトされた A1

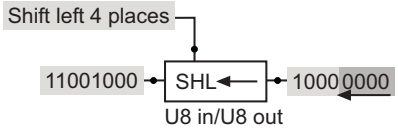
符号付きデータの最上位ビットはその値の符号を示すため、本コンポーネントは符号なしデータ型のみで使用します。

有効接続

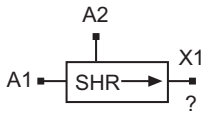
ピン	データ型	ピン	データ型
A1	UINT	X1	UINT
A2	UINT		

例: Shift Left





Shift Right（右シフト）



用途

- 整数信号のビットデータをシフトします。
- CAN データでのビットシフトに有用。

機能

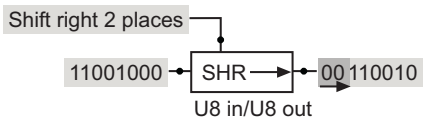
- X1 = A2 の値が示す位置数だけ右にシフトされた A1

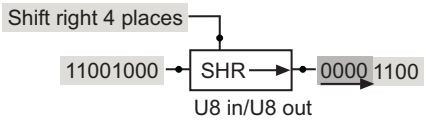
符号付きデータの最上位ビットはその値の符号を示すため、本コンポーネントは符号なしデータ型のみと使用します。

有効接続

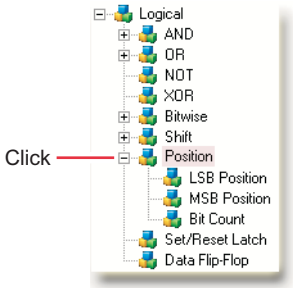
ピン	データ型	ピン	データ型
A1	UINT	X1	UINT
A2	UINT		

例: Shift Right





Position（位置）



LSB Position（最下位ビット位置）

用途

- 整数信号内で 1 に設定された最下位ビット（LSB）の位置を出力します。
- アラーム状態がバイトにコード化されている場合には、最優先アラームを示すことができます。

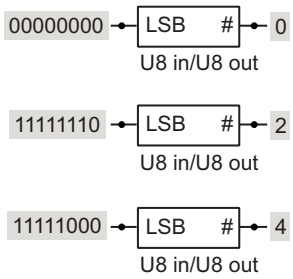
機能

- X1 = 1 の値を持つ最下位ビットの位置（右からの）
- 1 に設定されているビットがない場合には X1 = 0

有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	UINT

例: LSB Position



MSB Position（最上位ビット位置）

用途

- 整数信号内で 1 に設定された最上位ビット（MSB）の位置を出力します。
- アラーム状態がバイトにコード化されている場合には、最優先アラームを示すことができます。

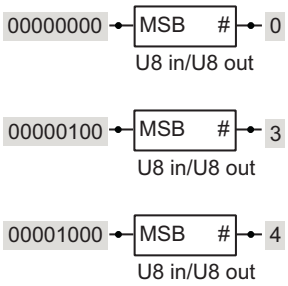
機能

- X1 = 1 の値を持つ最上位ビットの位置（右からの）
- 1 に設定されているビットがない場合には X1 = 0

有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	UINT

例: MSB Position



Bit Count（ビット数）

用途

- 整数信号内で True (1)に設定されたビットの数を出力します。
- アラーム状態がバイトにコード化されている場合には、アクティブアラームの数を示すことができます。

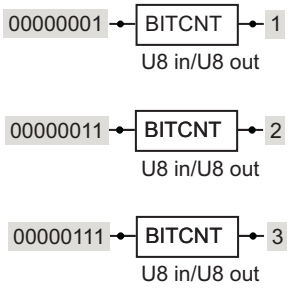
機能

- X1 = 1 の値を持つビットの数

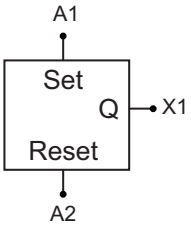
有効接続

ピン	データ型	ピン	データ型
A1	UINT	X1	UINT

例: Bit Count



Set/Reset Latch（セット/リセットラッチ）



用途

- ブール信号出力を保持します。
- Set を T にすると Q を T にラッチします（Reset が F のとき）。Q は Reset が T になるまで T のまま留まります。
- Reset が Set に優先します。

機能

- A1 = F および A2 = F であれば X1 = 無変化
- A1 = T および A2 = F であれば X1 = T にセット
- 以下の場合には X1 = F にリセット:
  - A1 = F および A2 = T
  - A1 = T および A2 = T
- X1 = T であれば A2 = T になるまで T を保持
- A2 = T であれば X1 = F

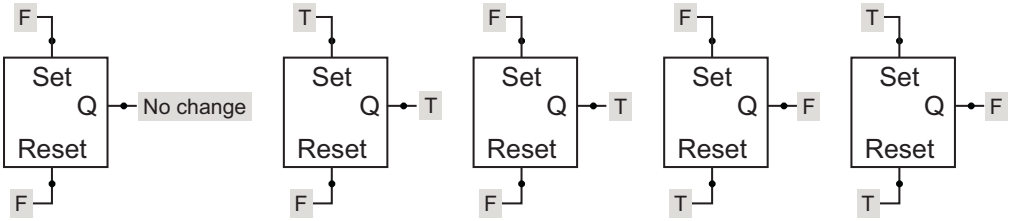
真理値表

A1 入力	A2 入力	X1 出力
F	F	無変化
T	F	T
F	T	F
T	T	F

有効接続

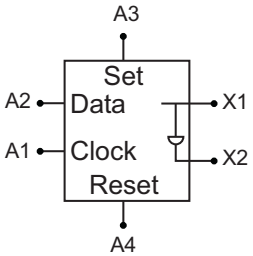
ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	BOOL		

例: Set/Reset Latch



Reset has priority over **Set**

Data Flip-Flop（データフリップフロップ）



用途

- ブール信号出力を保持します。
- ブール信号入力の遷移でブール信号出力をトグルします。
- X2 出力を A2 入力に接続することにより、トグル押しボタンとして使用できます。
- 常にリセット入力が優先されます。

機能

- A1、A2、A3、A4 = F であれば X1、X2 = 無変化
- A3 = T、A4 = F であれば X1 = T、X2 = F
- 以下の場合には X1 = F、X2 = T
  - A3 = F、A4 = T
  - A3 = T、A4 = T
- A1 = F から T への遷移、A2 = F、A3 = F、A4 = F であれば X1 = F、X2 = T
- A1 = F から T への遷移、A2 = T、A3 = F、A4 = F であれば X1 = T、X2 = F

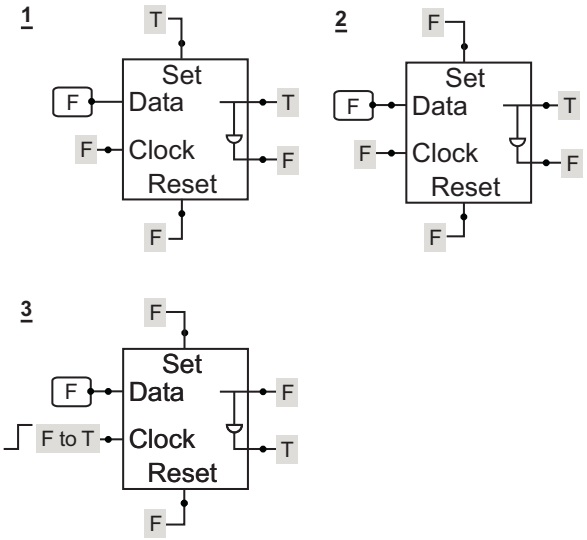
真理値表

A1 入力	A2 入力	A3 入力	A4 入力	X1 出力	X2 出力
F	F	F	F	変化なし	変化なし
——	——	T	F	T	F
——	——	F	T	F	T
——	——	T	T	F	T
F-T 遷移	F	F	F	変化なし	変化なし
T-F 遷移	T	F	F	トグル	トグル

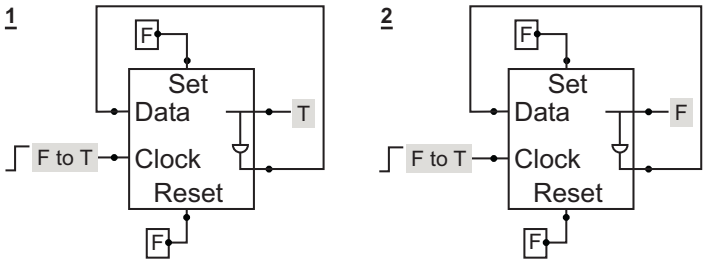
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	BOOL	X2	BOOL
A3	BOOL		
A4	BOOL		

例 1: Data Flip-Flop

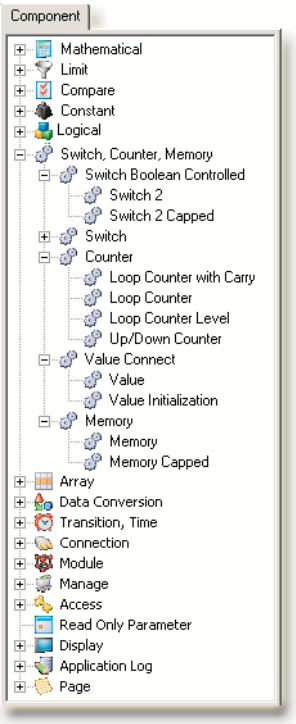


例 2: Data Flip-Flop

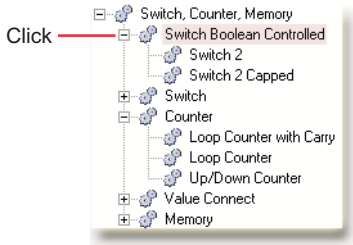


A1—Push-button input. Each button push toggles the X1 state.

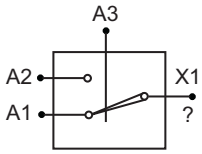
Switch, Counter, Memory（スイッチ、カウンタ、メモリ）



Switch Boolean Controlled（ブール制御スイッチ）



Switch 2（スイッチ 2）



用途

- 2 つの入力の間で出力を切り替えます。

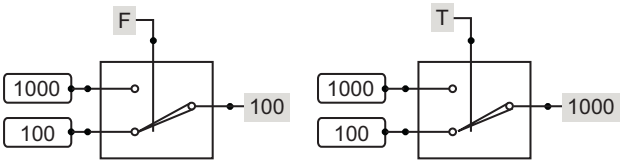
機能

- A3 = F であれば X1 = A1
- A3 = T であれば X1 = A2

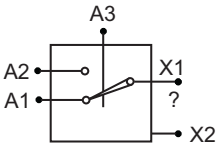
有効接続

ピン	データ型	ピン	データ型
A1, A2	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, LANG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, LANG
A3	BOOL		

例: Switch 2



Switch 2 Capped（上限付きスイッチ 2）



用途

- 2 つの入力の間で出力を切り替えます。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

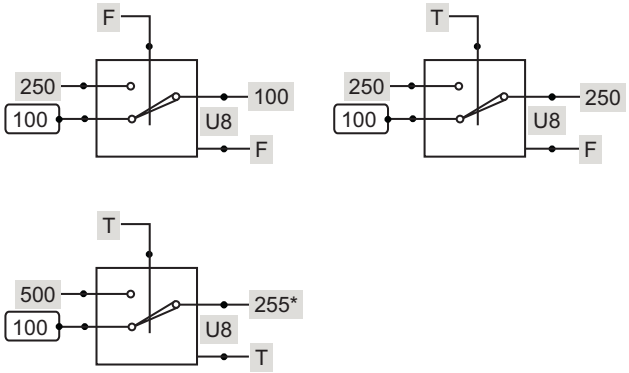
機能

- A3 = F であれば X1 = A1
- A3 = T であれば X1 = A2
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします

有効接続

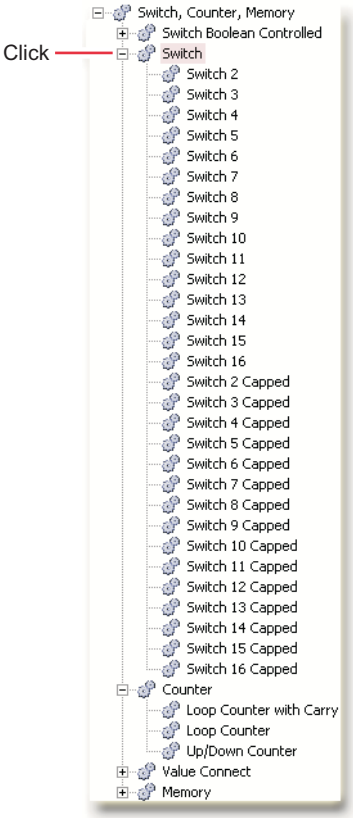
ピン	データ型	ピン	データ型
A1, A2	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, LANG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, LANG
A3	BOOL	X2	BOOL

例: Switch 2 Capped



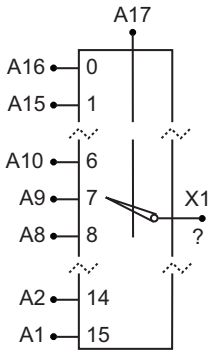
*255—Max value of the U8 data type

Switch（スイッチ）


Click
Switch, Counter, Memory
Switch Boolean Controlled
Switch
Switch 2
Switch 3
Switch 4
Switch 5
Switch 6
Switch 7
Switch 8
Switch 9
Switch 10
Switch 11
Switch 12
Switch 13
Switch 14
Switch 15
Switch 16
Switch 2 Capped
Switch 3 Capped
Switch 4 Capped
Switch 5 Capped
Switch 6 Capped
Switch 7 Capped
Switch 8 Capped
Switch 9 Capped
Switch 10 Capped
Switch 11 Capped
Switch 12 Capped
Switch 13 Capped
Switch 14 Capped
Switch 15 Capped
Switch 16 Capped
Counter
Loop Counter with Carry
Loop Counter
Up/Down Counter
Value Connect
Memory

Switch 2–16（スイッチ 2～16）

出力数の違いを除いて、スイッチ 2～スイッチ 16 コンポーネントは同様に機能します。ここではスイッチ 16 コンポーネントのみについて説明します。



用途

- 1～16 の入力の間で出力を切り替えます。

機能

- X1 = A(16 - A17 入力値)
- このコンポーネントの入力をフローティングのままにはできません。全ての入力はなんらかの値に接続する必要があります。

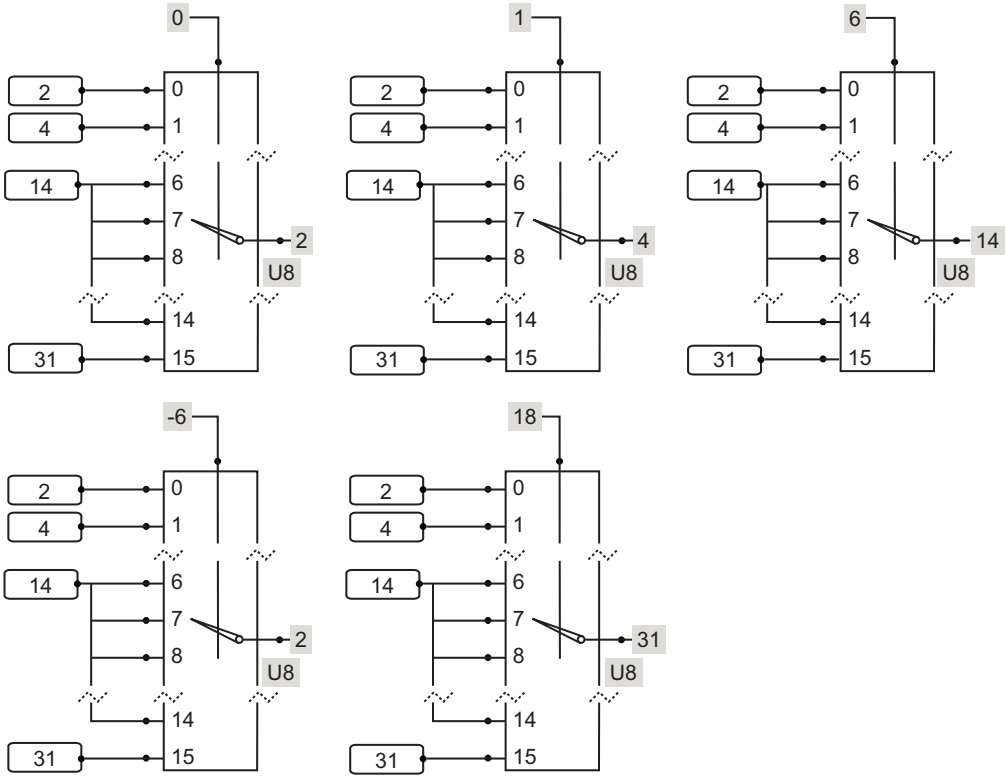
A17 入力　X1出力

A17 入力	X1 出力		A17 入力	X1 出力		A17 入力	X1 出力
< 0	A16		5	A11		11	A5
0	A16		6	A10		12	A4
1	A15		7	A9		13	A3
2	A14		8	A8		14	A2
3	A13		9	A7		15	A1
4	A12		10	A6		>15	A1

有効接続

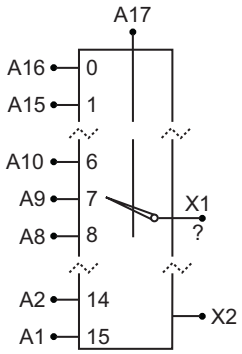
ピン	データ型	ピン	データ型
A1–A16	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, LANG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, LANG
A17	INT		

例: Switch 2–16



Switch 2–16 Capped（上限付きスイッチ 2～16）

出力数の違いを除いて、上限付きスイッチ 2～上限付きスイッチ 16 コンポーネントは同様に機能します。ここでは上限付きスイッチ 16 コンポーネントのみについて説明します。



用途

- 1～16 の入力の間で出力を切り替えます。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

機能

- X1 = A(16 - A17 入力値)
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします
- このコンポーネントの入力をフローティングのままにはできません。全ての入力はなんらかの値に接続する必要があります。

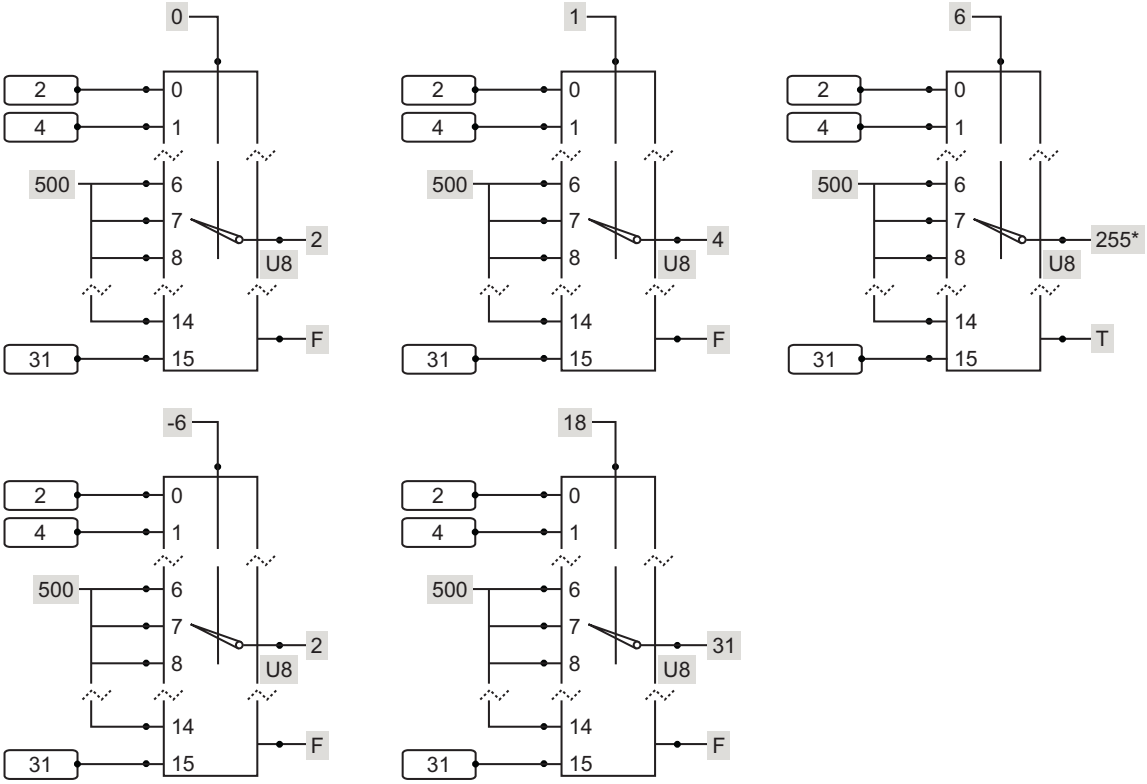
A17 入力　X 1出力

A17 入力	X1 出力		A17 入力	X1 出力		A17 入力	X1 出力
< 0	A16		5	A11		11	A5
0	A16		6	A10		12	A4
1	A15		7	A9		13	A3
2	A14		8	A8		14	A2
3	A13		9	A7		15	A1
4	A12		10	A6		>15	A1

有効接続

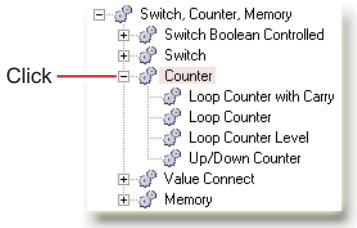
ピン	データ型	ピン	データ型
A1–A16	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG
A17	INT	X2	BOOL

例: Switch 2–16 Capped

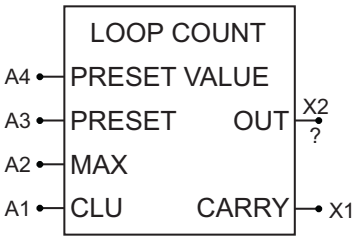


*255—Max value for the U8 data type

Counter（カウンタ）



Loop Counter with Carry（桁上げ付きループカウンタ）



用途

- 多数計数用のデイジーチェーン
- 動作時間の秒、分、時、日の合計に使用

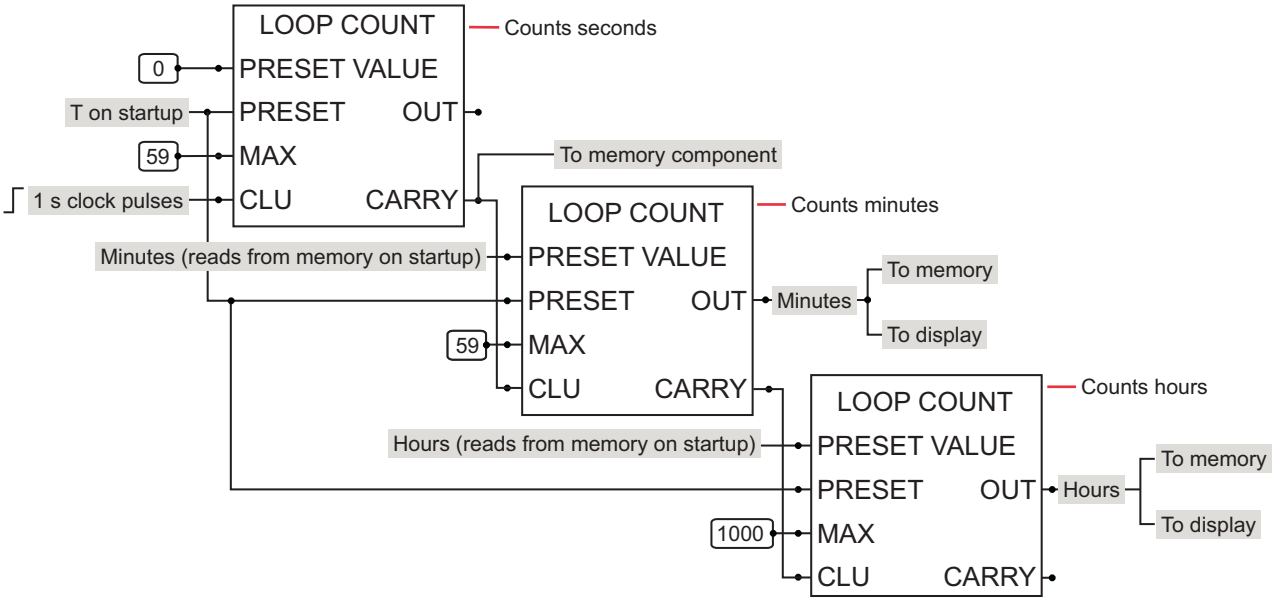
機能

- A1 = F-T 遷移が X2 を 1 ずつインクリメント
- A2 = 最大ループ値
- A3—
  - A3 = T であれば X2 = A4 (A3 が F になるまで A1 の遷移は X2 をインクリメントしません)
  - A3 = F であれば A1 の遷移が X2 をインクリメント
- A4 = プリセット値
- X1:
  - X2 = A2 で、A1 の F-T 遷移の 1 カウント後に T になります
  - X2 ≠ A4 + 1 である限り F のままです
- X2:
  - カウント出力
  - X1 = A2 で、A1 の F から T への遷移時に0にリセットします

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	UINT	X2	UINT
A3	BOOL		
A4	UINT		

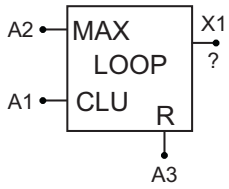
例: Loop Counter with Carry



この例は、秒、分、時を合計するタイマを示しています。

- 60 秒毎に、**LOOP COUNT** 秒コンポーネントは、以下のような True **CARRY** を出力します。
  - **LOOP COUNT** 分コンポーネントを 1 ずつインクリメントします。
  - **LOOP COUNT** 分コンポーネントと **LOOP COUNT** 時コンポーネントから **OUT** 値をメモリに書き込みます。
- 60 分毎に、**LOOP COUNT** 分コンポーネントは、True **CARRY** を出力します。この出力は、**LOOP COUNT** 時コンポーネントを 1 ずつインクリメントします。
- スタートアップ時に、True **PRESET** は以下を行います。
  - **PRESET VALUE 0** を **LOOP COUN** 秒コンポーネントに適用します。
  - メモリから読み取った **PRESET VALUE** を **LOOP COUNT** 分コンポーネントと **LOOP COUNT** 時コンポーネントに適用します。（本例が最初にダウンロードされるときには、**PRESET** 入力をすべて手動で **F** に初期化する必要があります。本例では、このロジックは示されていません。）

Loop Counter（ループカウンタ）



用途

- F→T 遷移をカウントして出力します

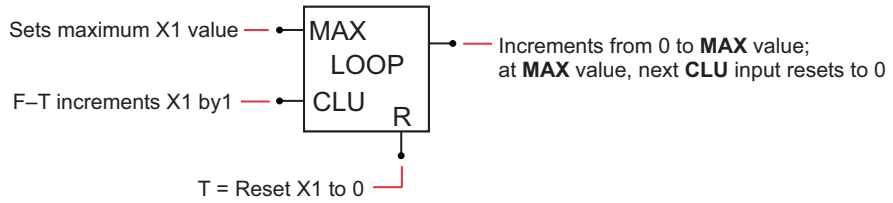
機能

- A1 = F→T 遷移ごとに X1 を 1 ずつインクリメントします
- A2 = X1 の最大値
- A3 = T であれば X1 = 0
- X1 は
  - 0 と A2 最大値の間の値です
  - A1 が F から T に遷移する毎にインクリメントします
  - X1 = A2 で、A1 が F から T への遷移する時に X1 が 0 にリセットします
  - A3 = T であれば 0 になります
  - A1 = F および A3 = F であれば変化しません

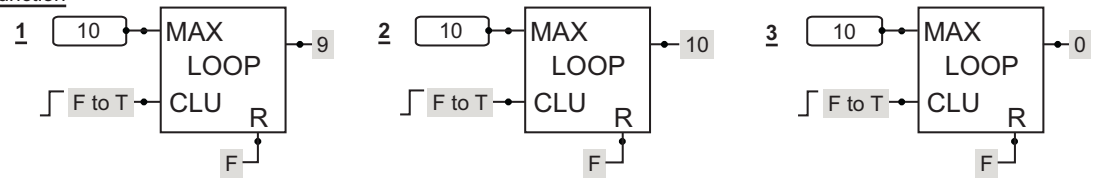
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	UINT
A2	UINT		
A3	BOOL		

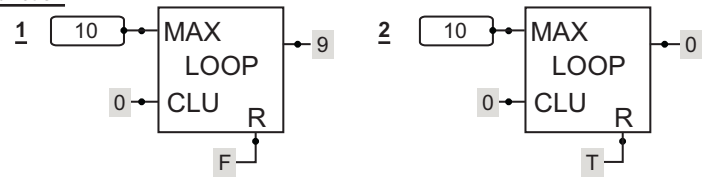
例: Loop Counter



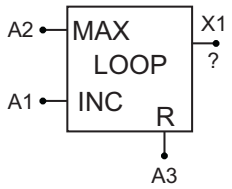
Count function



Reset function



Loop Counter Level



用途

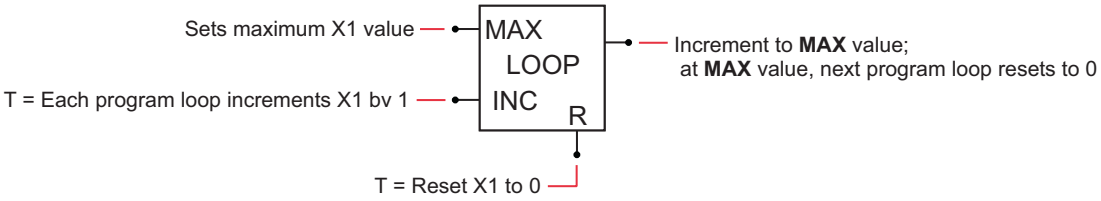
- OS 実行時間を元にしたカウント値を出力します Outputs a count based on OS execution time.

機能

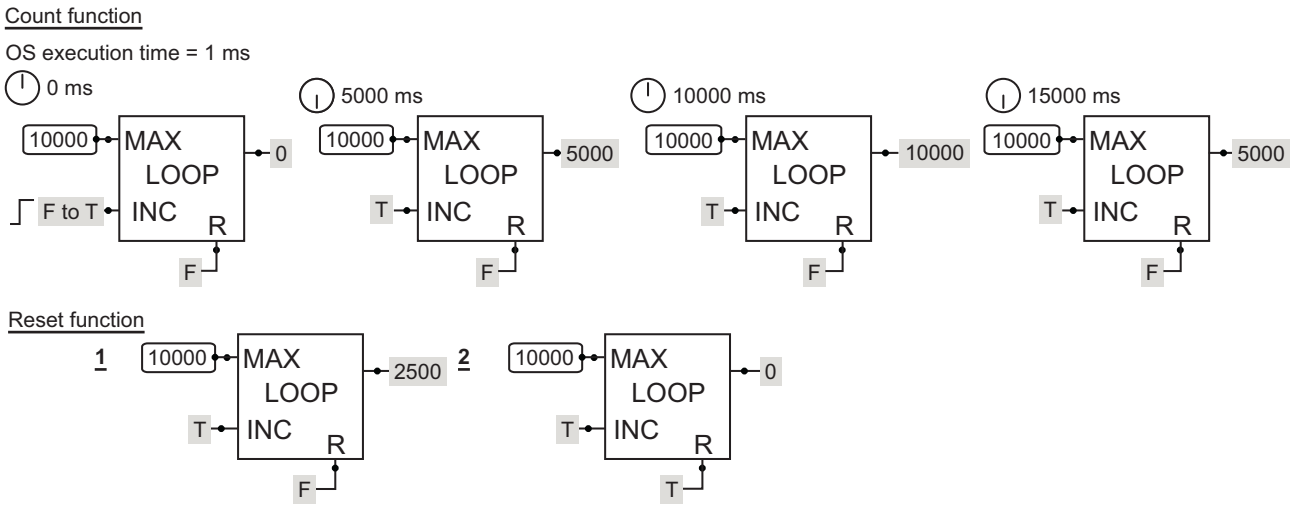
- A1 = T の場合、各プログラムループが完了するごとに X1 を 1 ずつインクリメントします
- A2 = X1 の最大値
- A3 = T であれば X1 = 0
- X1 は
  - 0 と A2 最大値の間の値です
  - X1 = A2 かつ A1 = T の時に X1 が 0 にリセットします
  - A3 = T であれば 0 になります
  - A1 = F かつ A3 = F であれば変化しません

有効接続

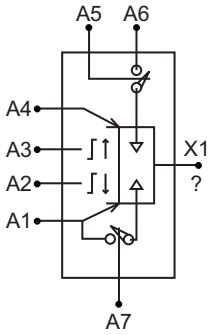
ピン	データ型	ピン	データ型
A1	BOOL	X1	UINT
A2	UINT		
A3	BOOL		



例: Loop Counter Level



Up/Down Counter（アップ/ダウンカウンタ）



用途

- ブール信号の遷移をカウントします。

機能

- X1 は A1 最小値から A4 最大値までの値
- A7 = T であれば X1 = A1
- A5 = T および A7 = F であれば X1 = A6
- A3 が F から T に遷移する毎に、X1 は A4 最大値になるまでインクリメントします
- A2 が F から T に遷移する毎に、X1 は A1 最小値になるまでデクリメントします
- A6 プリセットカウント値が A4 最大値よりも大きい場合には、X1 = A4 となります
- A6 プリセットカウント値が A1 最小値よりも小さい場合には、X1 = A1 となります
- A7 (最小値にセット) = T は A5 (プリセットイネーブル) = T に優先します

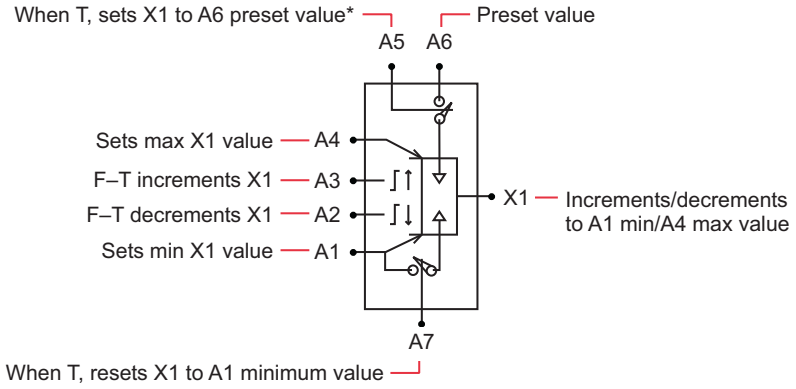
ピン機能

ピン	機能	ピン	機能
A1	最小カウント値	A5	プリセットカウントイネーブル
A2	デクリメントカウント	A6	プリセットカウント値
A3	インクリメントカウント	A7	セット最小カウント値
A4	最大カウント値	X1	カウント出力

有効接続

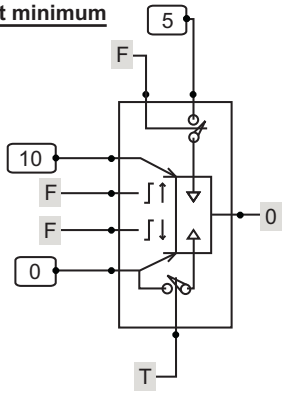
ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	BOOL		
A3	BOOL		
A4	INT		
A5	BOOL		
A6	INT		
A7	BOOL		

例: Up/Down Counter

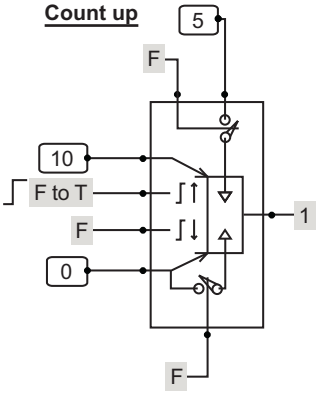


*If A5 is T:
· Sets X1 to A1 min value when A6 value is less than A1 value.
· Sets X1 to A4 max value when A6 value is greater than A4 value.

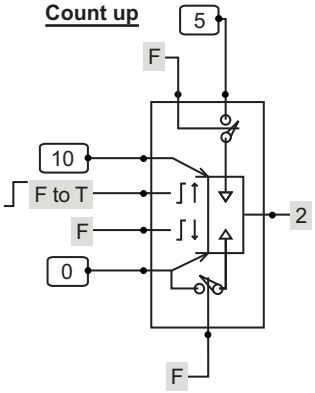
Set minimum



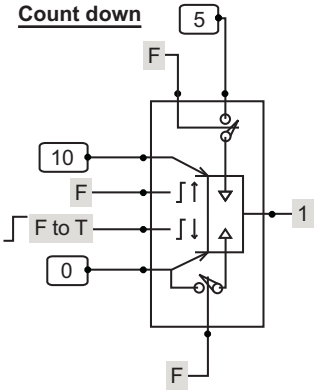
Count up



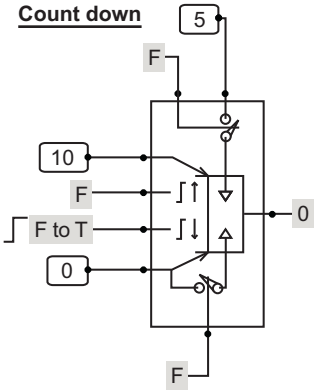
Count up



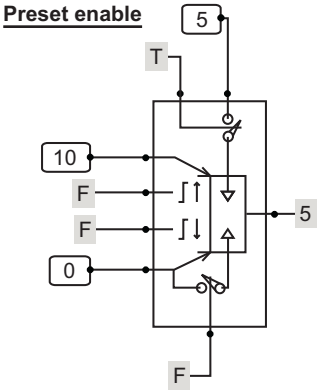
Count down



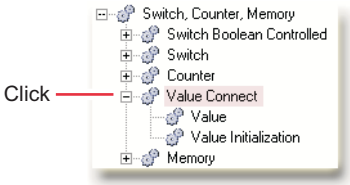
Count down



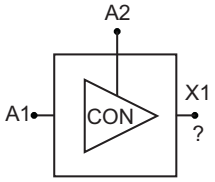
Preset enable



Value Connect（値の接続）



Value（値）



用途

- A2 = T であれば A1 = X1
- A2 = F であれば X1 の値は変化しません
- グループでの使用時に信号を切り替えます（例参照）。
- 値をサンプリングして保持します。
- これらのコンポーネントの X1 出力を互いにワイヤリングして、論理和関数を作ります。

機能

- 複数のコンポーネントを使用する場合、A2 入力がアクティブ（A2 = T）であるいずれか 1 つのコンポーネントの入力が出力に接続されます
- 2 つ以上の A2 入力がアクティブ（A2 = T）である場合は、実行順序が最後のものの値が有効となります

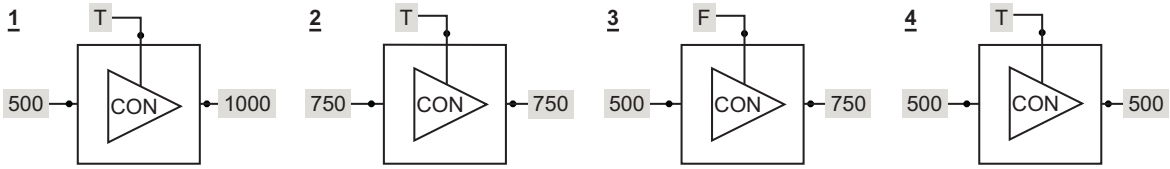
真理値表

A2 入力	X1 出力
F	無変化
T	A1

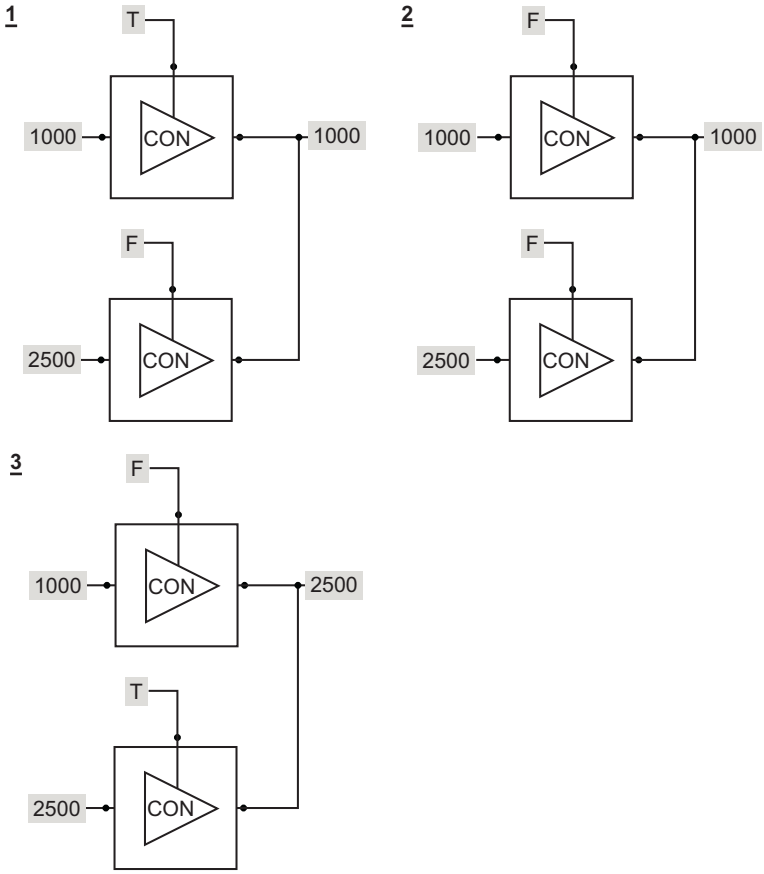
有効接続

ピン	データ型	ピン	データ型
A1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG
A2	BOOL		

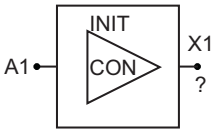
例 1: Value



例 2: Value



Value Initialization（値の初期化）



用途

- 各プログラムループの開始時に値を初期化します。

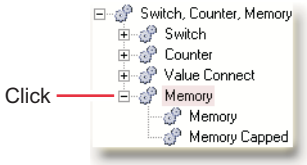
機能

- 各プログラムループの開始時に A1 = X1
- アプリケーションでの位置に関係なく、常に最初に実行します

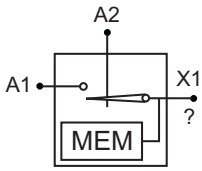
有効接続

ピン	データ型	ピン	データ型
A1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG

Memory（メモリ）



Memory（メモリ）



用途

- 信号変化後に信号値をサンプリングしてその値を保持します。

機能

- A2 = T であれば X1 = A1
- A2 = F であれば X1 = X1
- 本コンポーネントからの複数の出力を互いにワイヤリングすることはできません
- 複数の出力を互いにワイヤリングする必要がある場合は、**Value**コンポーネント（***210***ページ***Value***コンポーネント参照）を使用します

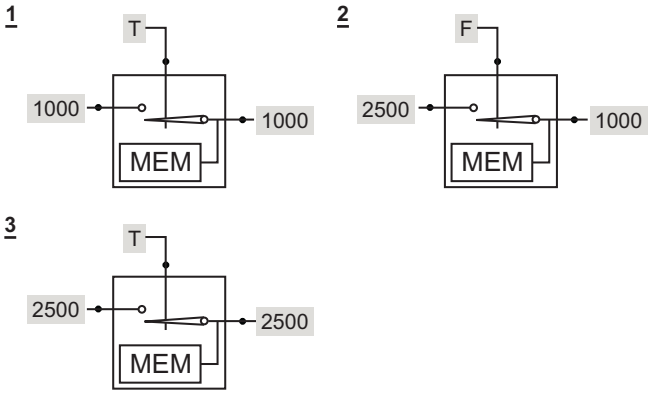
真理値表

A2 入力	X1 出力
F	無変化
T	A1

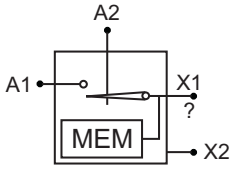
有効接続

ピン	データ型	ピン	データ型
A1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG
A2	BOOL		

例: Memory



Memory Capped（上限付きメモリ）



用途

- 信号変化後に信号値をサンプリングしてその値を保持します。
- オーバーフローする場合には X1 出力をクランプし、X2 出力を T に設定してオーバーフロー状態を示します。
- 上限付きコンポーネントのブール出力を互いにワイヤリングすることができ、オーバーフロー状態がネットで T を出力します。

機能

- A2 = T であれば X1 = A1
- A2 = F であれば X1 = X1
- X1 がオーバーフローしない場合には、X2 = F となります
- X1 がオーバーフローする場合には、以下のようになります。
  - X2 = T
  - X1 = そのデータ型の最小値または最大値でクランプします
  - X2 は各プログラムループの開始時に F にリセットします
- 本コンポーネントからの複数の X1 出力を互いにワイヤリングすることはできません
- 複数の出力を互いにワイヤリングする必要がある場合は、**Value**コンポーネント（***210***ページ参照）を使用します。

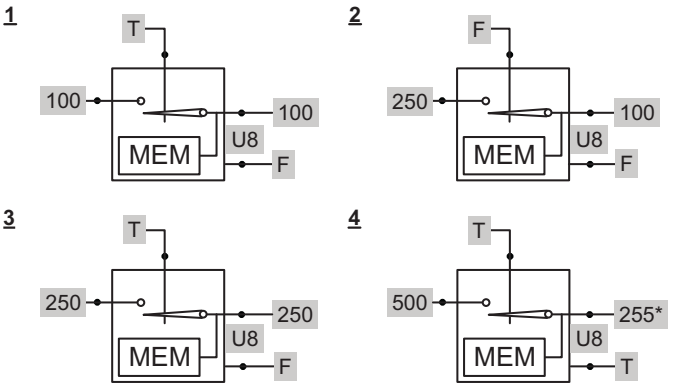
有効接続

ピン	データ型	ピン	データ型
A1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG
A2	BOOL	X2	BOOL

真理値表

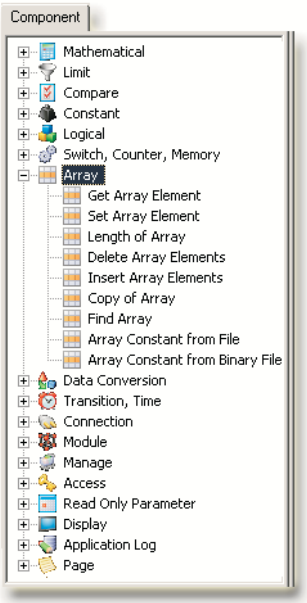
A2 入力	X1 出力
F	無変化
T	A1

例: Memory Capped

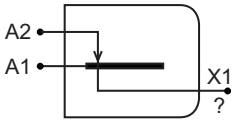


*255—Max value for the U8 data type

Array（配列）


Component
Mathematical
Limit
Compare
Constant
Logical
Switch, Counter, Memory
Array
Get Array Element
Set Array Element
Length of Array
Delete Array Elements
Insert Array Elements
Copy of Array
Find Array
Array Constant from File
Array Constant from Binary File
Data Conversion
Transition, Time
Connection
Module
Manage
Access
Read Only Parameter
Display
Application Log
Page

Get Array Element（配列要素の取得）



用途

- 配列内の要素の値を出力します。
- A1 入力が U8 かつ X1 出力が S8 に型指定されていると正常に動作しません。回避策：X1 出力を S8 に型指定してください。

機能

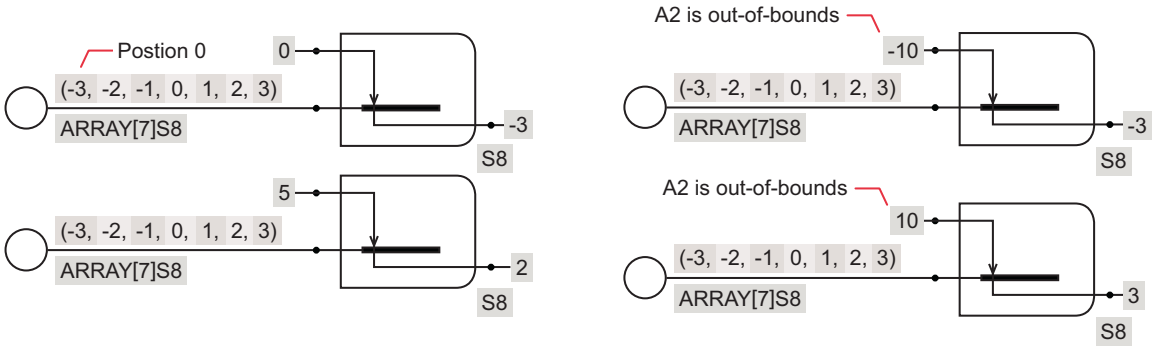
- A1 = 配列
- A2 = 出力する要素の位置

  配列内の最初の要素は位置 0 にあります
- X1 = コピーされた要素
- A2 がマイナス側に範囲外のとき、X1= 位置 0 の配列要素
- A2 がプラス側に範囲外のとき、X1= 最後の配列要素
- 配列は最大 32,767 の要素を含むことができます
- 入出力ピンの ARRAY データ型が異なっている場合には、X1 出力にオーバーフローがもたらされることがあります
- 配列データ型の詳細に関しては、72 ページの「配列データ型」を参照。

有効接続

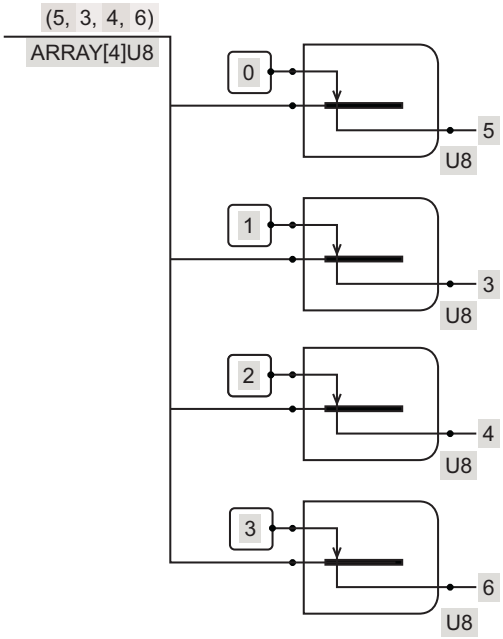
ピン	データ型	ピン	データ型
A1	ARRAY	X1	BOOL, INT
A2	U8, S8, S16		

例 1—Get Array Element

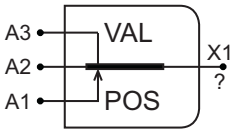


例 2: Get Array Element

以下の例には、**Get Array Element** コンポーネントを用いて、配列から個々の要素を抜き出す方法が示されています。



Set Array Element（配列要素の設定）



用途

- 配列内の要素を定義値に設定します。

機能

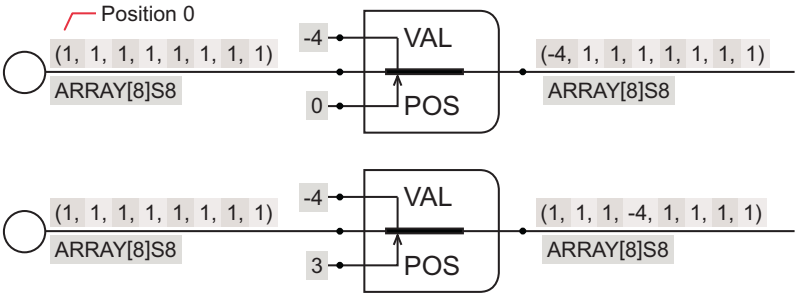
- A1 = 定義する要素の位置

  配列内の最初の要素は位置 0 にあります
- A2 = 設定する要素のある配列
- A3 = 設定する要素の値
- X1 = 変更した要素のある配列
- A1（設定位置）がプラス側あるいはマイナス側に範囲を超えている場合、X1 = A2
- 配列は最大 32,767 の要素を含むことができます
- 入出力ピンの ARRAY データ型が異なっている場合には、X1 出力にオーバーフローがもたらされることがあります
- 配列データ型の詳細に関しては、72 ページの「配列データ型」を参照してください。

有効接続

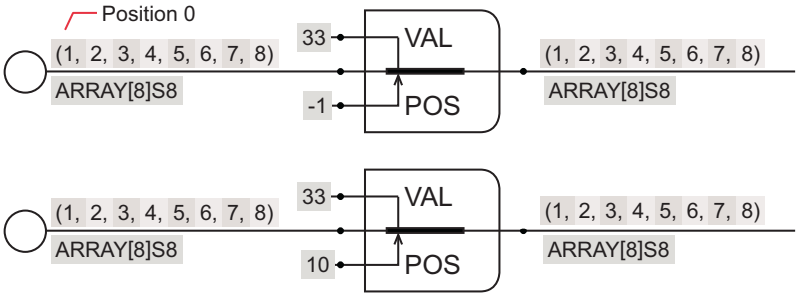
ピン	データ型	ピン	データ型
A1	U8, S8, S16	X1	ARRAY
A2	ARRAY		
A3	INT, BOOL		

例 1: Set Array Element



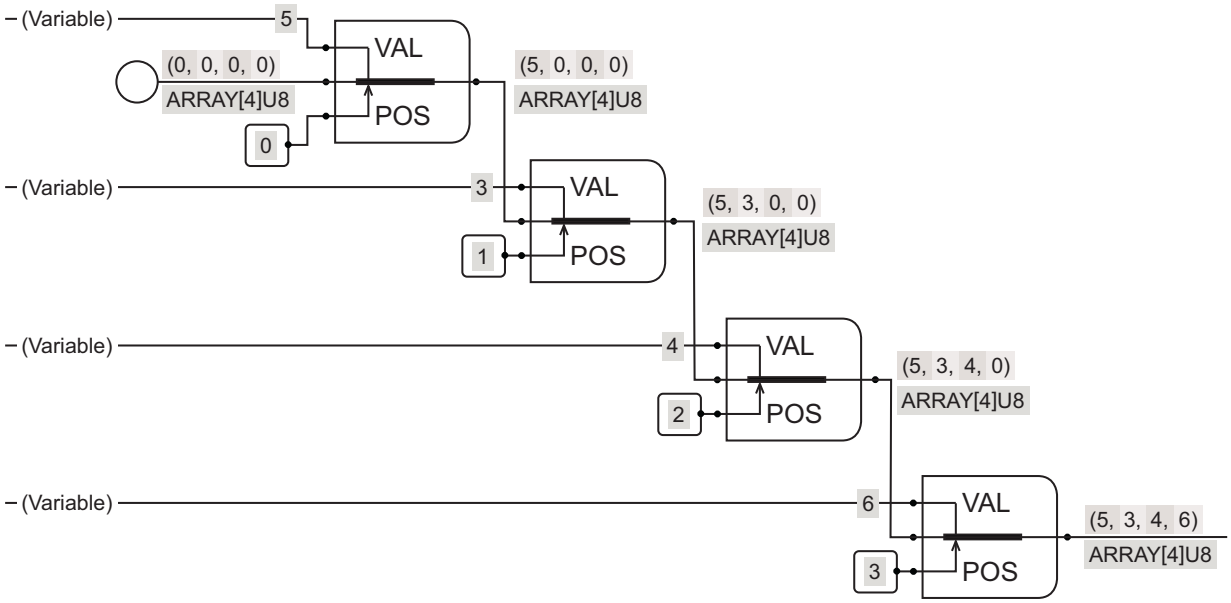
例 2: Set Array Element

以下の例は **A1 Pos** 入力が範囲外の場合に何が起こるかを示しています。

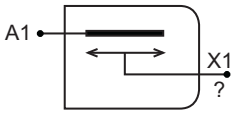


例 3: Set Array Element

以下の例には、Set Array Element コンポーネントを用いて、変数を含む配列を構成する方法が示されています。



Length of Array（配列の長さ）



用途

- 配列内の要素の数を出力します。

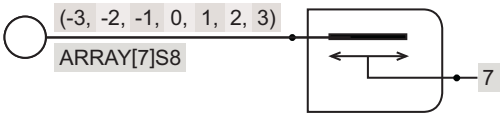
機能

- A1 = 配列
- X1 = 配列内の要素の数
- 配列は最大 32,767 の要素を含むことができます
- 入出力ピンの ARRAY データ型が異なっている場合には、X1 出力にオーバーフローがもたらされることがあります
- 配列データ型の詳細に関しては、72 ページの「配列データ型」を参照してください。

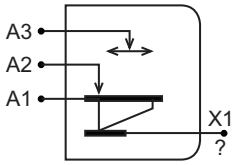
有効接続

ピン	データ型	ピン	データ型
A1	ARRAY	X1	UINT

例: Length of Array



Delete Array Elements（配列要素の削除）



用途

- 配列から要素を削除します。

機能

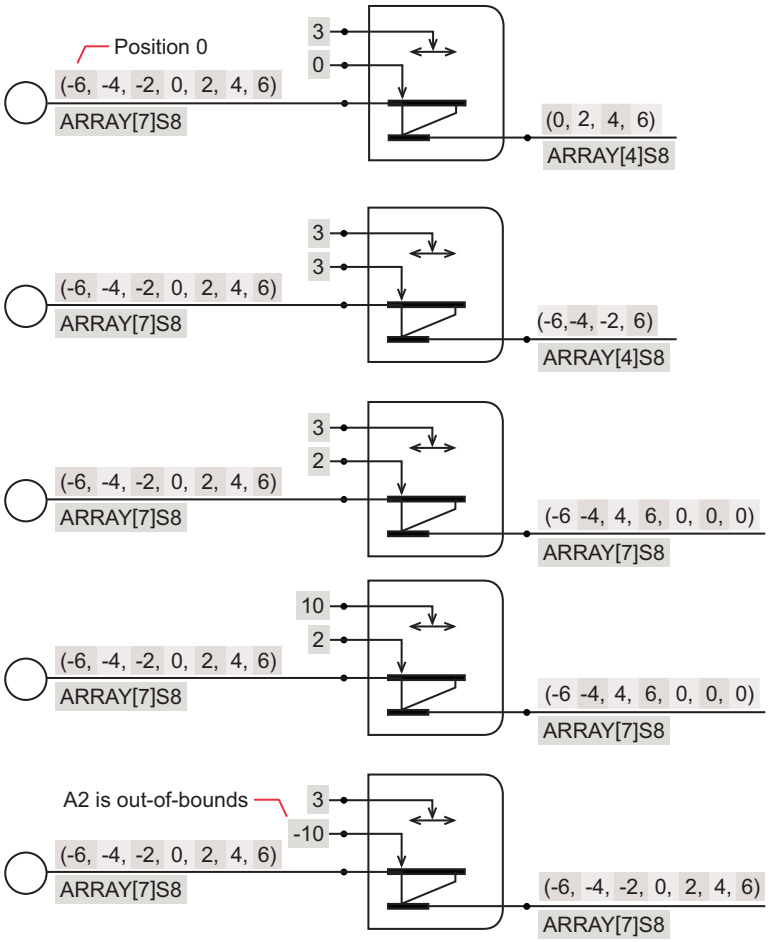
- A1 = 要素を削除する配列
- A2 = 削除する最初の要素の開始位置

  配列内の最初の要素は位置 0 にあります
- A3 = 配列から削除する要素の数
- X1 = 要素を削除した後の配列
- A2（削除する最初の位置）がプラス側あるいはマイナス側に範囲外の場合、X1 = A1
- 配列は最大 32,767 の要素を含むことができます
- 入出力ピンの ARRAY データ型が異なっている場合には、X1 出力にオーバーフローがもたらされることがあります
- 入力配列と出力配列は同じデータ型でなければなりません
- 配列データ型の詳細に関しては、72 ページの「配列データ型」を参照してください。

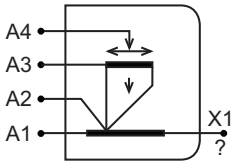
有効接続

ピン	データ型	ピン	データ型
A1	ARRAY	X1	ARRAY
A2	U8, S8, S16		
A3	INT		

例: Delete Array Elements



Insert Array Elements（配列要素の挿入）



用途

- 追加の配列要素を配列に挿入します。

機能

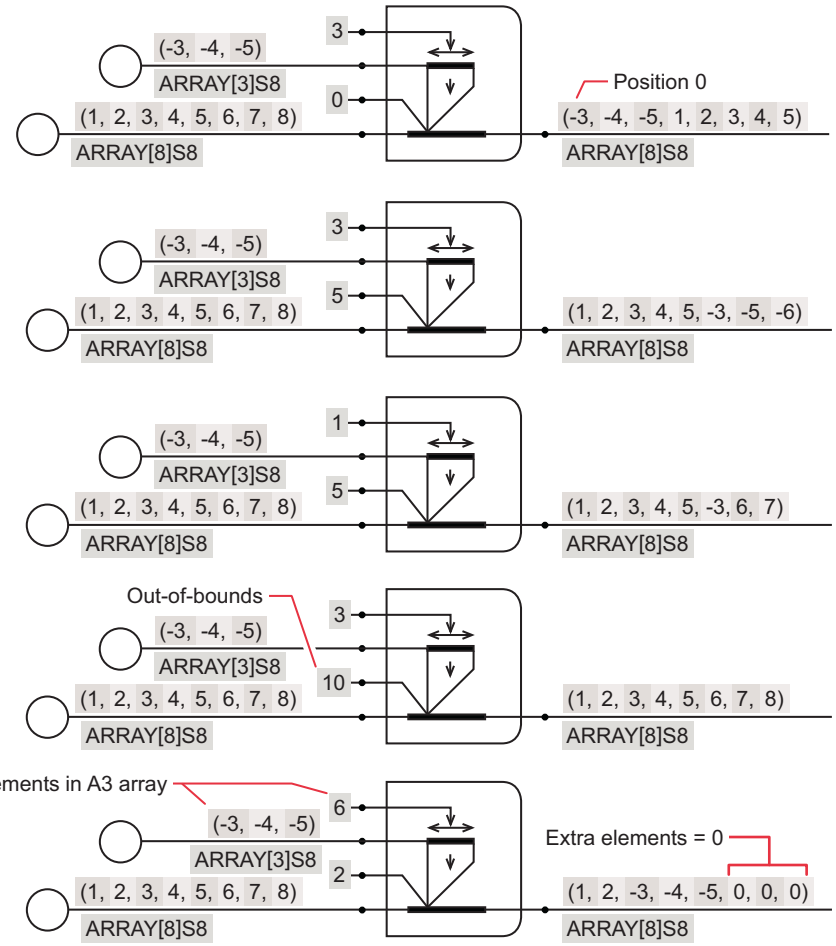
- A1 = 新しい要素を挿入する配列
- A2 = 新しい要素を挿入する開始位置

  配列内の最初の要素は位置 0 にあります
- A3 = 挿入する要素のある配列
- A4 = 挿入する要素の数
- X1 = 要素を挿入した後の配列
- A2（挿入位置）が範囲外の場合、X1 = A1
- A4（挿入する要素の数）> A3 配列の要素の数の場合、A4 の残りの要素はゼロが出力されます
- 配列は最大 32,767 の要素を含むことができます
- 入出力ピンの ARRAY データ型が異なっている場合には、X1 出力にオーバーフローがもたらされることがあります
- 入力配列と出力配列は同じデータ型でなければなりません
- 配列データ型の詳細に関しては、72 ページの「配列データ型」を参照してください。

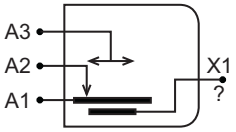
有効接続

ピン	データ型	ピン	データ型
A1	ARRAY	X1	ARRAY
A2	U8, S8 S16		
A3	ARRAY		
A4	UINT		

例: Insert Array Elements



Copy of Array（配列のコピー）



用途

- 配列から要素をコピーして出力します。

機能

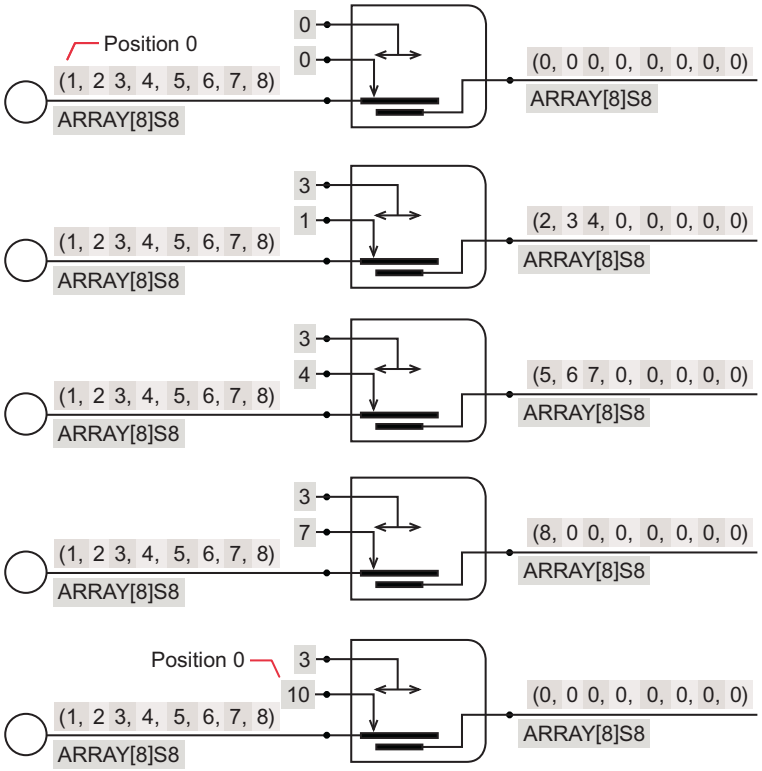
- A1 = コピーする要素のある配列
- A2 = コピーする最初の要素の位置

  配列内の最初の要素は位置 0 にあります
- A3 = コピーする要素の数
- X1 = コピーされた要素
- 配列は最大 32,767 の要素を含むことができます
- 入出力ピンの ARRAY データ型が異なっている場合には、X1 出力にオーバーフローがもたらされることがあります
- 配列データ型の詳細に関しては、72 ページの「配列データ型」を参照してください。

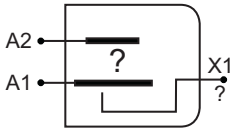
有効接続

ピン	データ型	ピン	データ型
A1	ARRAY	X1	ARRAY
A2	U8, S8, S16		
A3	UINT		

例: Copy of Array



Find Array（配列の検索）



用途

- マッチする要素の位置またはマッチする複数の要素の開始位置を検索します。

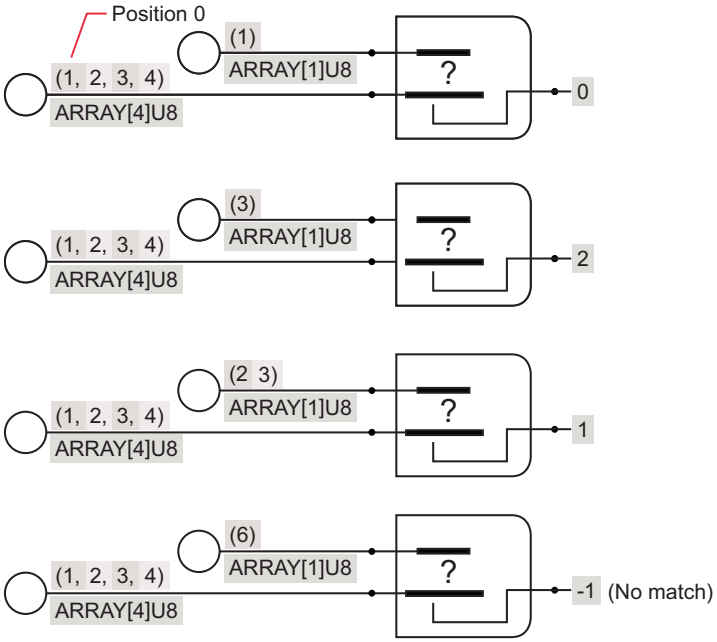
機能

- A1 = マッチする要素のある配列
- A2 = マッチする要素
- X1 = マッチする要素の位置またはマッチする複数の要素の開始位置
- マッチしない場合には、X1 = -1 となります
- 配列データ型の詳細に関しては、72 ページの「配列データ型」を参照してください。

有効接続

ピン	データ型	ピン	データ型
A1	ARRAY	X1	S16
A2	ARRAY		

例: Find Array



Array Constant from File（ファイルからの配列定数）

FILE • ? (File name)
X1
? (Data type)

用途

- テキスト (***.txt) ファイルからデータを入力します。
- データはカンマ区切りでなければならず、テキストファイル毎に 1 つの配列のみ。

機能

- X1 = ファイルからのカンマ区切り値に設定された要素を持つ配列
- ? = テキストファイル名
  - ファイル名は大文字で始める必要があります

    正： Array1.txt

    誤： array1.txt
  - ファイル名にはピリオド(.)を含めることはできません

    正： Array1.txt

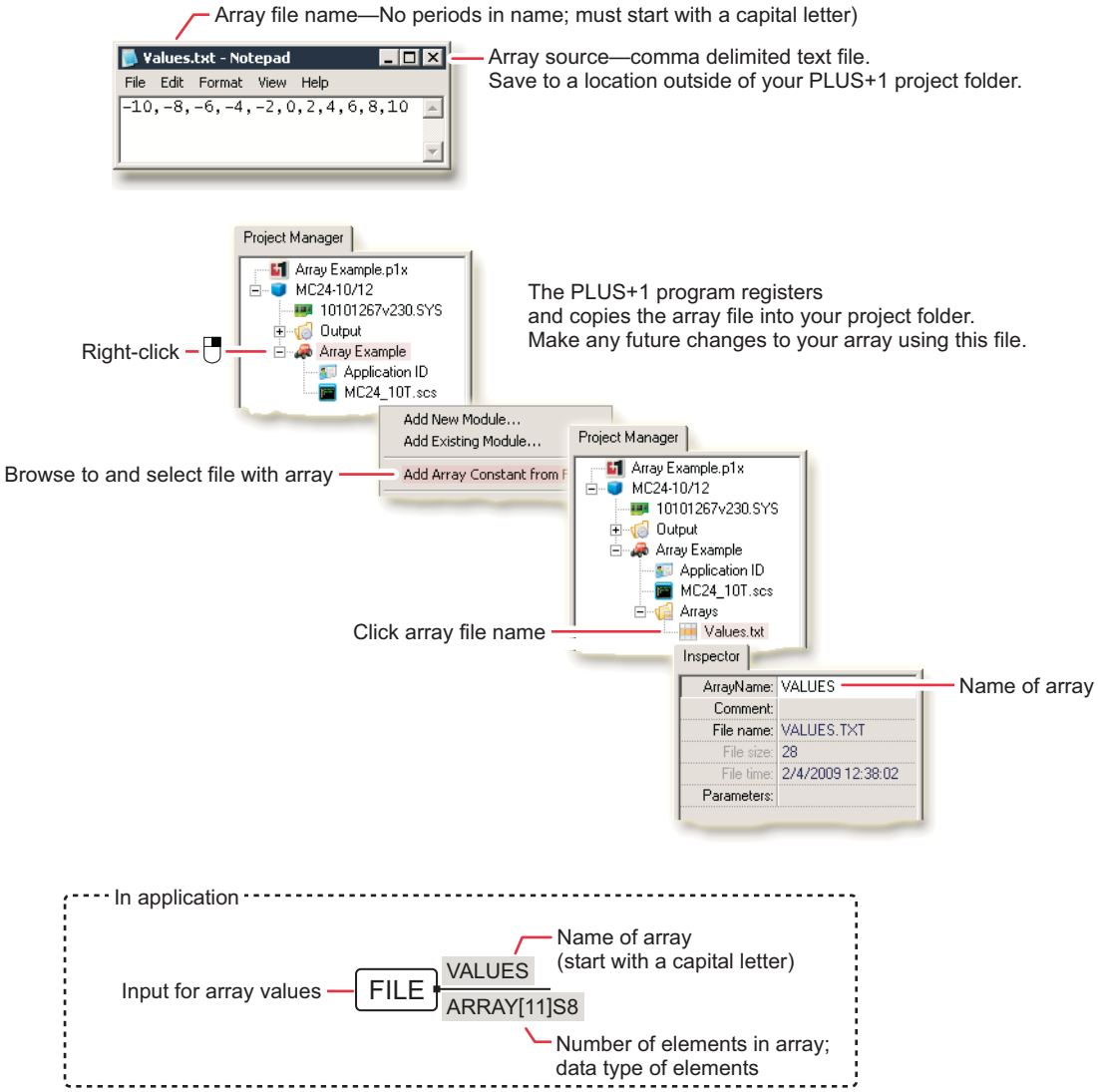
    誤： Array.1.txt

配列データ型の詳細に関しては、72 ページの「配列データ型」を参照してください。

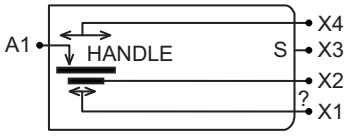
有効接続

ピン	データ型	ピン	データ型
—	—	X1	ARRAY

例: Array Constant from File



Array Constant from Binary File（バイナリファイルからの配列定数）



用途

- バイナリ(***.bin)ファイルからデータを入力します。

機能

- A1 = 配列内の最初の要素の開始位置
- X1 = 配列内の有効な要素の数を示す
- X2 = バイナリファイルからの配列データ
- X3 = 状態
- X4 = バイナリファイルの全体の長さを示す
- HANDLE = バイナリファイル名（HANDLE は分かりやすい名前に変更可能）
  - ファイル名は大文字で始める必要があります

    正： Binary1.bin

    誤： binary1.bin
  - ファイル名にはピリオド(.)を含めることはできません

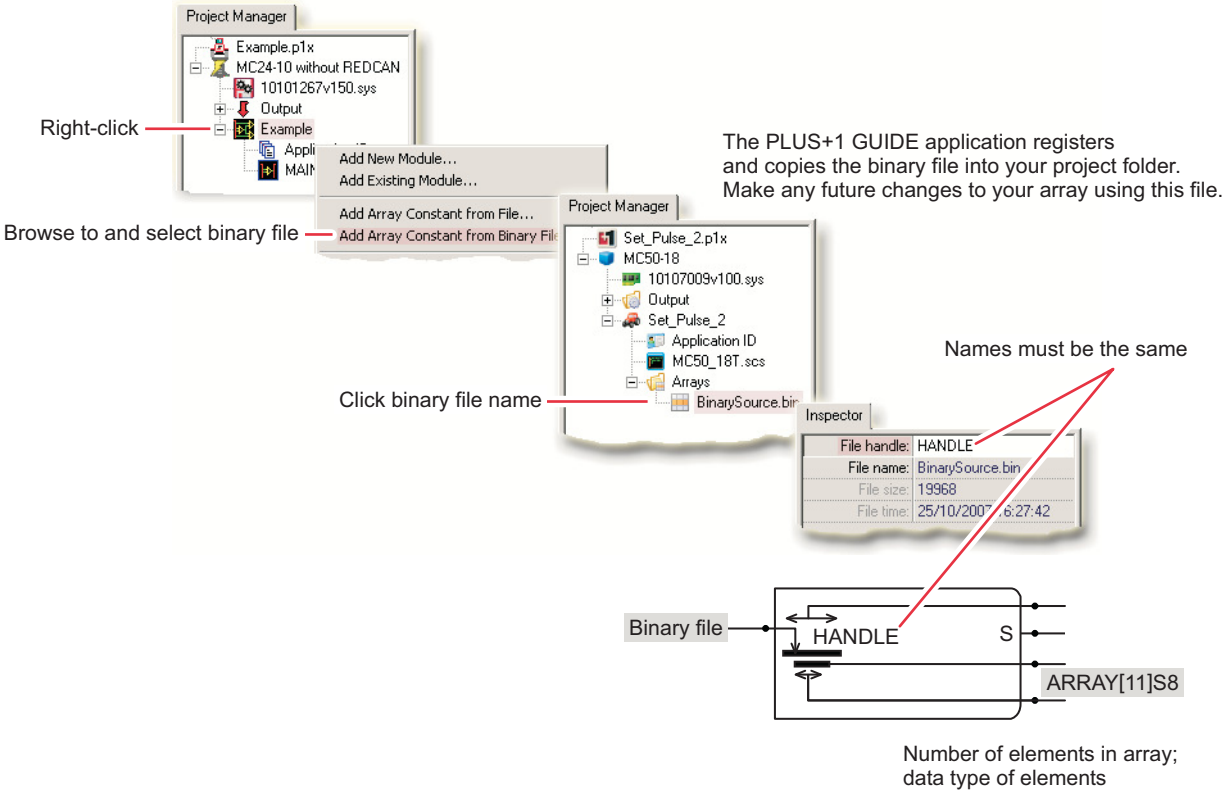
    正： Binary1.bin

    誤： Binary.1.bin

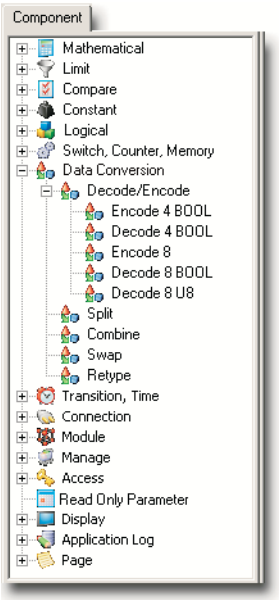
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
		X2	ARRAY
		X3	INT
		X4	INT

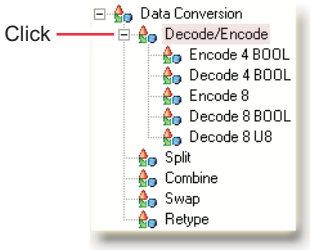
例: Array Constant from Binary File



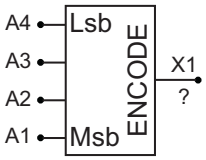
Data Conversion（データ変換）



Decode/Encode（復号化/符号化）



Encode 4 BOOL（4 ブール信号の符号化）



用途

- ブール信号を整数信号に符号化します。
- ブールデータを CAN メッセージにマージすることができます。

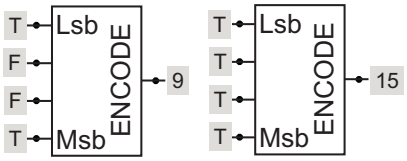
機能

- X1 出力において
  - A1 入力状態が bit 3 を設定します
  - A4 入力状態が bit 0 を設定します

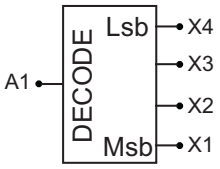
有効接続

ピン	データ型	ピン	データ型
A1–A4	BOOL	X1	INT

例: Encode 4 BOOL



Decode 4 BOOL（4 ブール信号への復号化）



用途

- 整数信号を 4 つのブール信号に復号します。

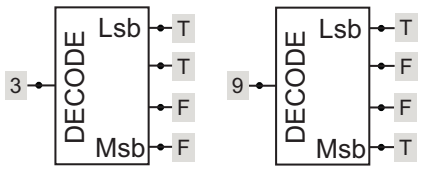
機能

- X1 = bit 3 の出力
- X4 = bit 0 の出力
- マイナス値の A1 は出力をすべてゼロに設定します

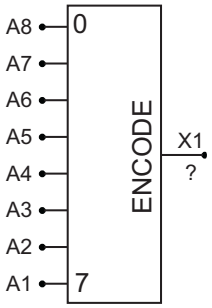
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1–X4	BOOL

例: Decode 4 BOOL



Encode 8（符号化 8）



用途

- ブール信号を整数信号、ARRAY[INT]信号、または ARRAY[BOOL]信号に符号化します。
- データを CAN データにマージすることができます。

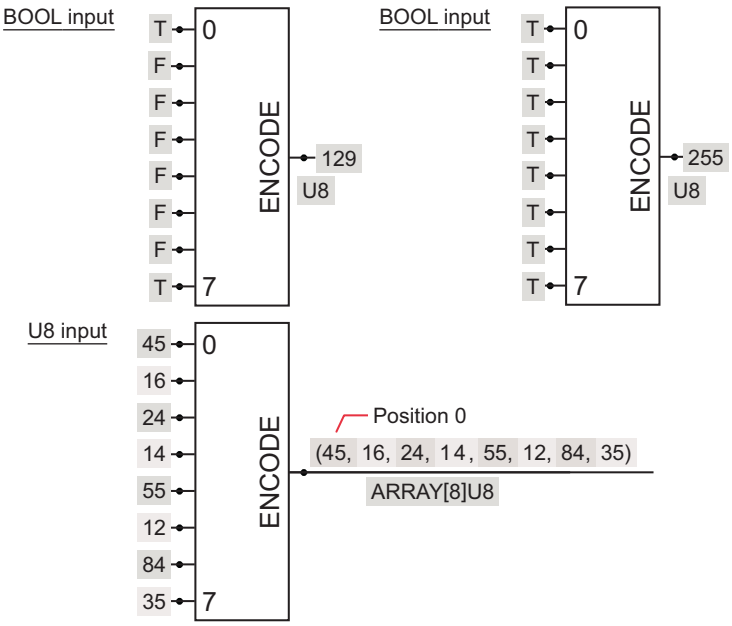
機能

- INT 出力の X1 に対して
  - A1 入力が bit 7 を設定します
  - A8 入力が bit 0 を設定します
- ARRAY 出力の X1 に対して
  - A1 入力が位置 7 の要素の値を設定します
  - A8 入力が位置 0 の要素の値を設定します

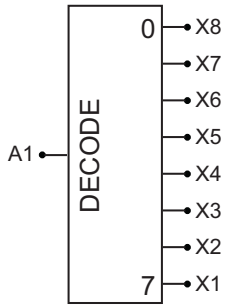
有効接続

ピン	データ型	ピン	データ型
A1–A8	BOOL, INT	X1	ARRAY[BOOL], ARRAY [INT], INT

例: Encode 8


BOOL input
T
F
F
F
F
F
F
T
0
ENCODE
7
129
U8
BOOL input
T
T
T
T
T
T
T
T
0
ENCODE
7
255
U8
U8 input
45
16
24
14
55
12
84
35
0
ENCODE
7
Position 0
(45, 16, 24, 14, 55, 12, 84, 35)
ARRAY[8]U8

Decode 8 BOOL（8 ブール信号への復号化）



用途

- 整数信号を 8 つのブール信号に復号します。
- CAN データバイトからデータを抜き出すことができます。

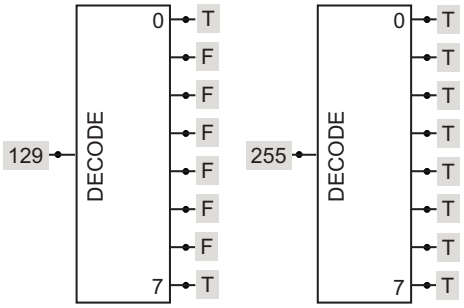
機能

- X1 = bit 7 の出力
- X8 = bit 0 の出力
- マイナス値の A1 は出力をすべてゼロに設定します

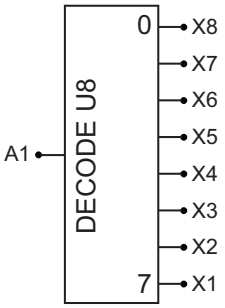
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1–X8	BOOL

例: Decode 8 BOOL



Decode 8 U8（8 つのU8 信号への復号化）



用途

- 8つの整数を含んだ要素を持つ配列を8つの U8 信号に複号します。
- CAN データフィールドからデータバイトを抜き出すことができます。

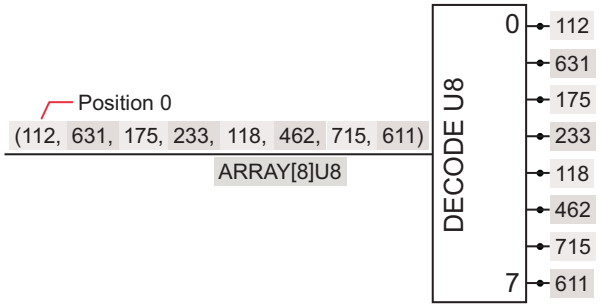
機能

- X1 = 位置 7 の要素の値を出力します
- X2-X7 = 位置 6～1 の要素の値を出力します
- X8 = 位置 0 の要素の値を出力します

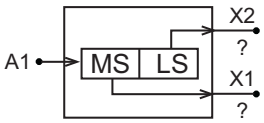
有効接続

ピン	データ型	ピン	データ型
A1	ARRAY[8]U8	X1–X8	U8

例: Decode 8 U8



Split（分割）



用途

- 整数信号をそれぞれ同じビット数を持つ2つの信号に分けます。
- 結合 CAN メッセージを2つに分けることができます。

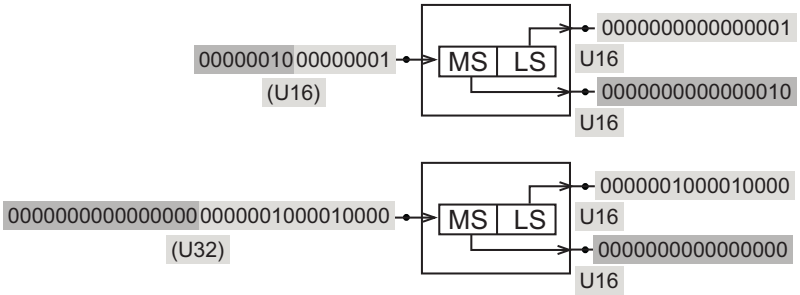
機能

- X1 = A1 入力のうち、上位ビットを含む半分
- X2 = A1 入力のうち、下位ビットを含む半分

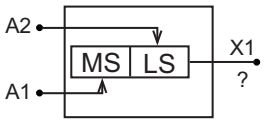
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
		X2	INT

例: Split



Combine（結合）



用途

- 同一サイズの2つの整数信号を結合して、ビット数が2倍の新たな信号にします。
- 2つの別個の受信 U8 CAN メッセージを結合して、1つの U16 信号にすることができます。

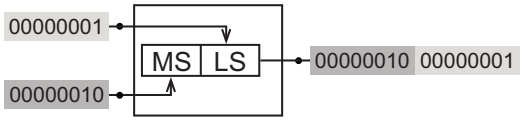
機能

- X1 = A2 の左にシフトされた A1
- 常に同一ビットサイズの入力を結合します
  - 8 ビット A1 入力と 8 ビット A2 入力が結合して、16 ビット X1 出力がもたらされます
  - 16 ビット A1 入力と 16 ビット A2 入力が結合して、32 ビット X1 出力がもたらされます

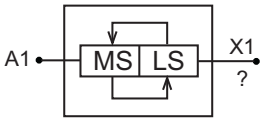
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT
A2	INT		

例: Combine



Swap（入れ替え）



用途

- 整数信号内のビットを入れ替えます。

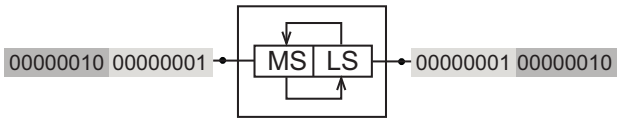
機能

- X1 = A1 入力内のビットの下位半分が A1 入力内のビットの上位半分と位置を入れ替えたもの
- 入れ替え後
  - 16 ビット A1 入力内の上位 8 ビットが X1 出力内の下位 8 ビットになります
  - 16 ビット A1 入力内の下位 8 ビットが X1 出力内の上位 8 ビットになります

有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT

例: Swap



Retype（型変更）

用途

- 信号のデータ型を変更します。

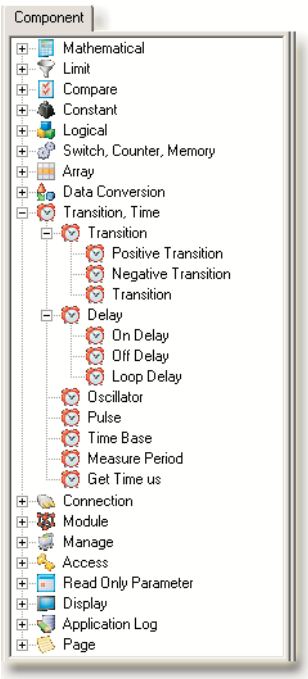
機能

- X1 = A1
- S8 から U8 や U16 から U8 等のデータ型の変更は、X1 出力をオーバーフローさせることがあります

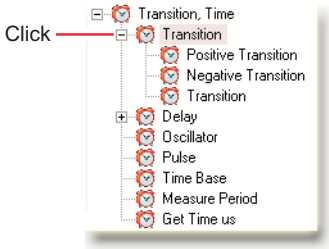
有効接続

ピン	データ型	ピン	データ型
A1	INT	X1	INT

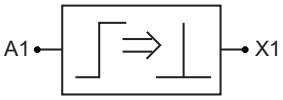
Transition, Time（遷移、時間）



Transition (遷移)



Positive Transition (立ち上がり遷移)



用途

- ブール信号の正遷移(F から T)を検出します。

機能

- A1 が F から T へ遷移した後、1プログラムループの間だけ X1 = T

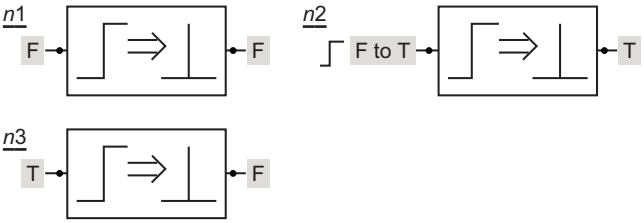
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL

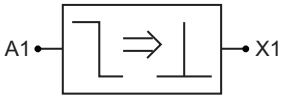
X1 出力

プログラムループ	A1	X1
*n*1	F	F
*n*2	T	T
*n*3	T	F

例: Positive Transition



Negative Transition（立ち下がり遷移）



用途

- ブール信号の負遷移（T から F）を検出します。

機能

- A1 が T から F へ遷移した後、1プログラムループの間だけ X1 = T

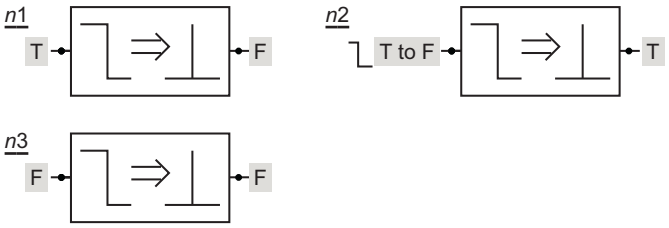
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL

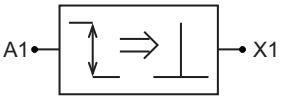
X1 出力

プログラムループ	A1	X1
*n*1	T	F
*n*2	F	T
*n*3	F	F

例: Negative Transition



Transition（遷移）



用途

- ブール信号の F から T または T から F の遷移を検出します。

機能

- A1 が F から T または T から F へ遷移した後、1プログラムループの間だけ X1 = T

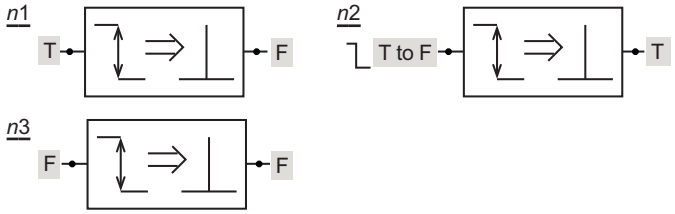
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL

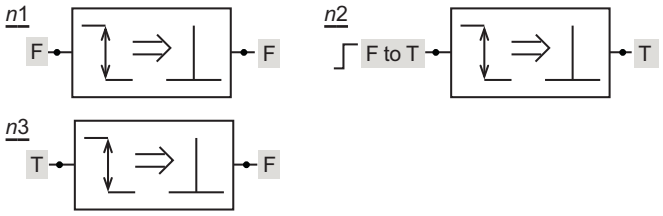
X1 出力

例1プログラムループ	A1	X1		例 2 プログラムループ	A1	X1
*n*1	T	F		*n*1	F	F
*n*2	F	T		*n*2	T	T
*n*3	F	F		*n*3	T	F

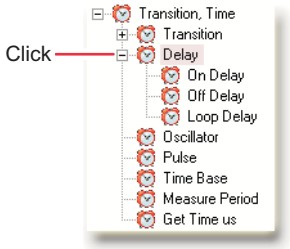
Example1—Transition



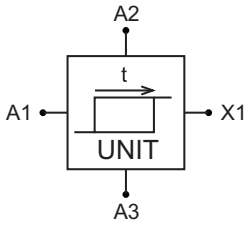
Example2—Transition



Delay（遅延）



On Delay（オン遅延）



用途

- ブール信号の F から T への復帰を指定された時間だけ遅らせます。

機能

- A1 = F であれば X1 = F
- A1 が F から T に移行すると、保持時間（A3 x A2）が開始
- 保持時間中は X1 = F
- 保持時間が終了すると X1 = T
- A3 = TLOOP のとき、遅延時間は A2 – 1

  （例：A3 が TLOOP かつ A2 が 2 の場合、遅延時間は 1 プログラムループ）

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	UINT		
A3	TIME		

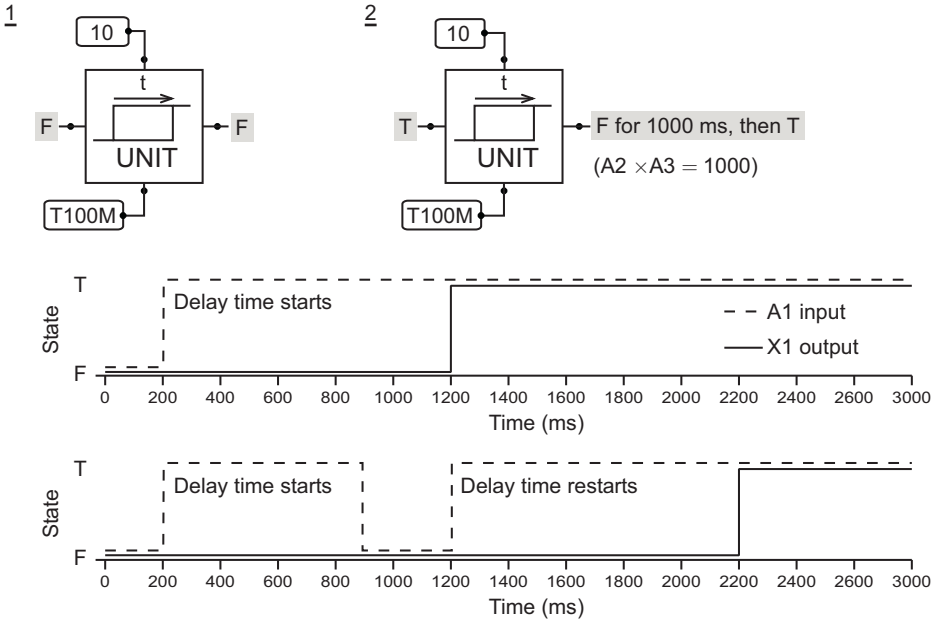
時間単位

Unit	値
T1M	1 ms
T10M	10 ms
T100M	100 ms

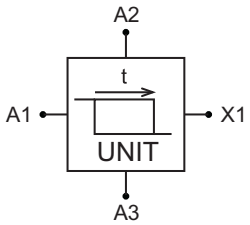
Unit	値
T1S	1 s
T60S	60 s
TIH	1 h

Unit	値
TLOOP	プログラムループ

例: On Delay


1
10
t
F
UNIT
F
T100M
2
10
t
T
UNIT
F for 1000 ms, then T
(A2 ×A3 = 1000)
T100M
State
T
F
Delay time starts
A1 input
X1 output
0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 3000
Time (ms)
State
T
F
Delay time starts
Delay time restarts
0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 3000
Time (ms)

Off Delay（オフ遅延）



用途

- ブール信号の T から F への復帰を設定時間だけ遅らせます。

機能

- A1 = T であれば X1 = T
- A1 が T から F に移行すると、保持時間（A3 x A2）が始まります
- 保持時間中は X1 = T
- 保持時間が終了すると X1 = F
- A3 = TLOOP のとき、遅延時間は A2 – 1

  （例：A3 が TLOOP かつ A2 が 2 の場合、遅延時間は 1 プログラムループ）

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	UINT		
A3	TIME		

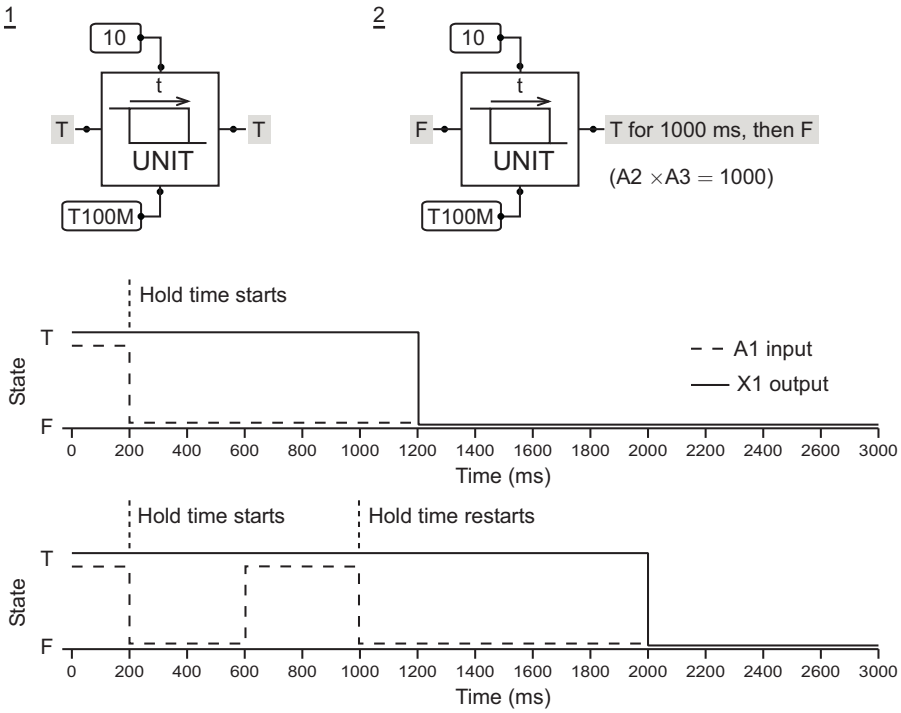
時間単位

Unit	値
T1M	1 ms
T10M	10 ms
T100M	100 ms

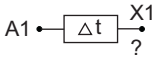
Unit	値
T1S	1 s
T60S	60 s
TIH	1 h

Unit	値
TLOOP	プログラムループ

例: Off Delay


1
10
t
T
T
UNIT
T100M
2
10
t
F
T for 1000 ms, then F
UNIT
(A2 ×A3 = 1000)
T100M
Hold time starts
State
T
F
A1 input
X1 output
0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 3000
Time (ms)
Hold time starts
Hold time restarts
State
T
F
0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 3000
Time (ms)

Loop Delay（ループ遅延）



用途

- 1コンポーネント実行の遅延の後に、A1 入力を X1 に出力します。

機能

- 1プログラムループの遅延の後に X1 = A1

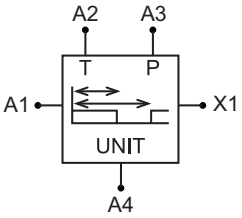
有効接続

ピン	データ型	ピン	データ型
A1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, OBJ	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, OBJ

***Loop Delay** コンポーネント入出力*

プログラムループ	A1 入力	X1 出力
n1	5	5
n2	7	5
n3	7	7
n4	7	7

Oscillator（発振器）



用途

- 交互に切り替わるブール信号を生成します。
- カウンタと結合してタイマとなります。

機能

- A2 はオン時間（コンポーネント内の T）
- A3 は周期（コンポーネント内の P）
- A2 ≥ A3 であれば X1 = T（注：これは通常の用法ではありません）
- A1 が F から T に移行すると、A2 値と A3 値がサンプリングされ、以下のようになります
  - 最初の保持時間（A2）中は X1 = T
  - 次の保持時間（A3－A2）中は X1 = F

  A1 = T の間、この機能は反復されます。反復中には、X1 が F から T に移行する毎に、A2 値と A3 値がサンプリングされ、保持時間が再計算されます。
- A2 > 0 であれば最短でも1プログラムループ中は X =T になります

有効接続

ピン	データ型	Pin	データ型
A1	BOOL	X1	BOOL
A2	UINT		
A3	UINT		
A4	TIME		

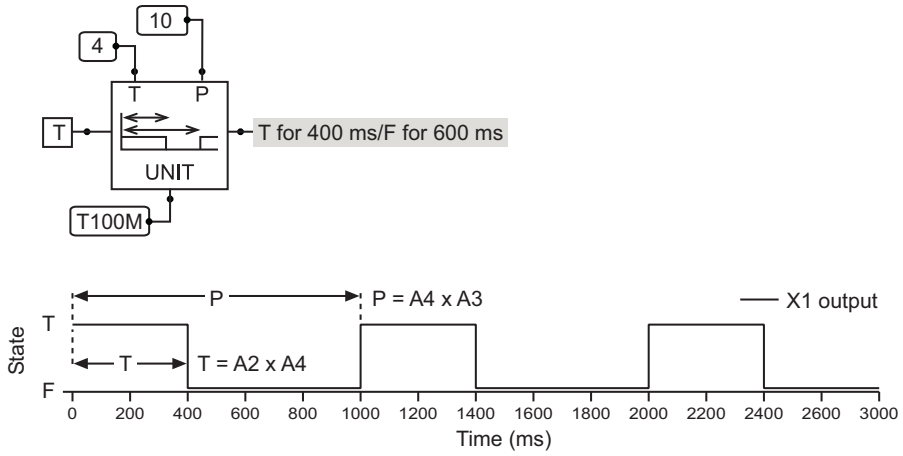
時間単位

Unit	値
T1M	1 ms
T10M	10 ms
T100M	100 ms

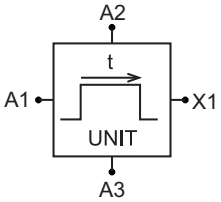
Unit	値
T1S	1 s
T60S	60 s
TIH	1 h

Unit	値
TLOOP	プログラムループ

例: Oscillator



Pulse (パルス)



用途

- ブール信号を指定した時間だけ T に保持したあと、F に戻ります。

機能

- A1 = F であれば X1= F
- A1 が F から T に移行すると、保持時間 (A3 × A2) が開始
- 保持時間中は X1 = T
- 保持時間が終了すると X1 = F

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	UINT		
A3	TIME		

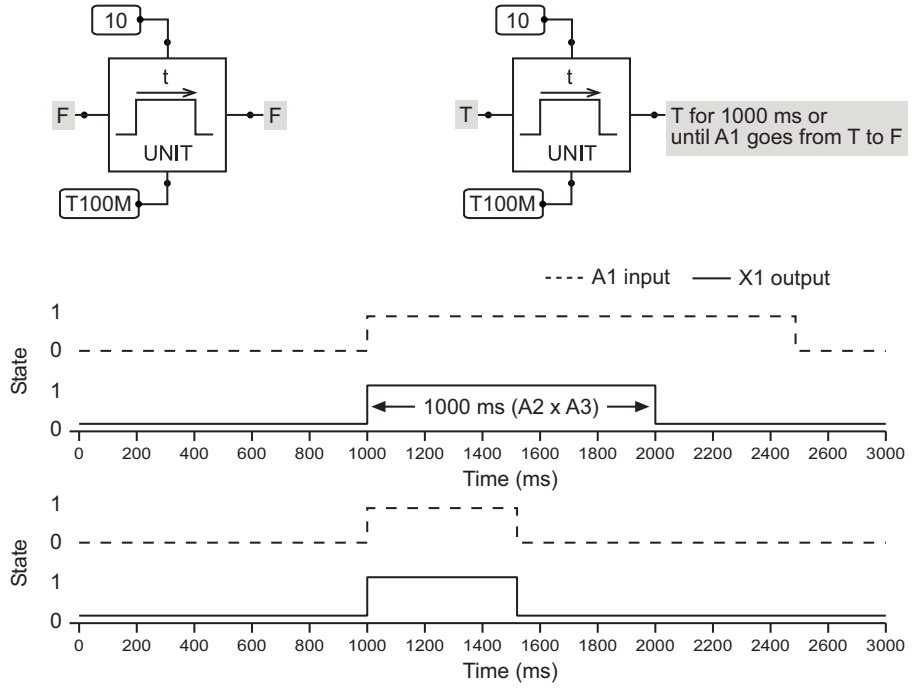
時間単位

Unit	値
T1M	1 ms
T10M	10 ms
T100M	100 ms

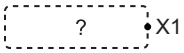
Unit	値
T1S	1 s
T60S	60 s
TIH	1 h

Unit	値
TLOOP	プログラムループ

例: Pulse


10
t
F
F
UNIT
T100M
10
t
T
T for 1000 ms or
until A1 goes from T to F
UNIT
T100M
A1 input
X1 output
State
1
0
1
0
1000 ms (A2 x A3)
0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 3000
Time (ms)
State
1
0
1
0
0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 3000
Time (ms)

Time Base（基準時間）



用途

- 基準時間定数を出力します。

機能

- X1 = ? で規定される基準時間の値

有効接続

ピン	データ型	ピン	データ型
—	—	X1	TIME

時間単位

Unit	値
T1M	1 ms
T10M	10 ms
T100M	100 ms

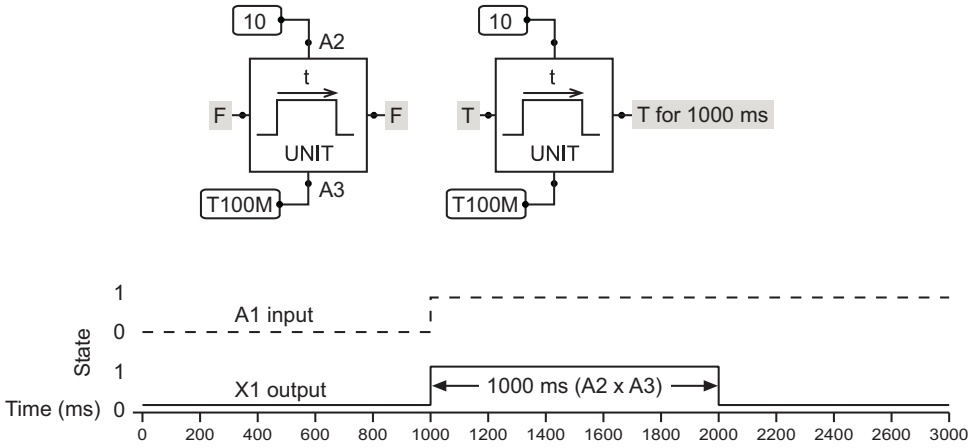
Unit	値
T1S	1 s
T60S	60 s
TIH	1 h

Unit	値
TLOOP	プログラムループ

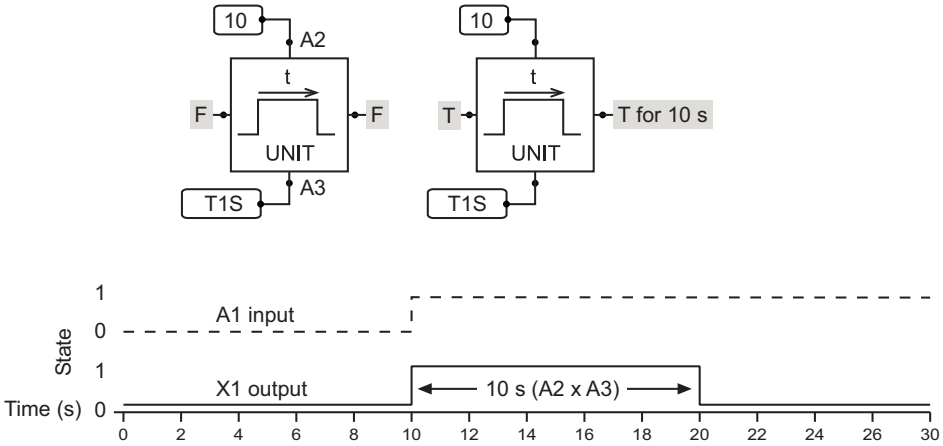
時間基準値が小さいほど、出力分解能が正確になります。分解能の詳細に関しては 71 ページを参照してください。

例: Time Base

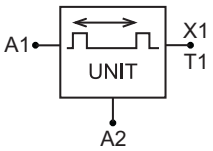
Pulse component with a 100 ms Time Base



Pulse component with a 1 s Time Base



Measure Period（周期計測）



用途

- 交互に切り替わる信号の周期を計測します。

機能

- X1 = A1 の最新の 2 つの F から T への遷移の間の時間
- X1 = A1 ÷ A2

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	UINT
A2	TIME		

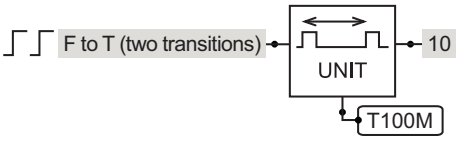
時間単位

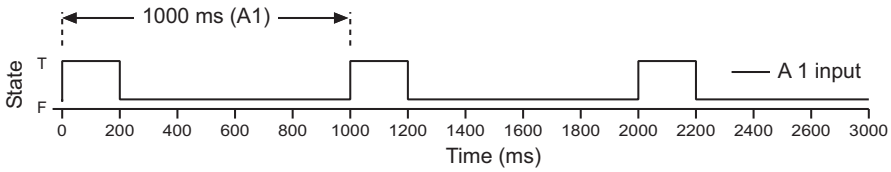
Unit	値
T1M	1 ms
T10M	10 ms
T100M	100 ms

Unit	値
T1S	1 s
T60S	60 s
TIH	1 h

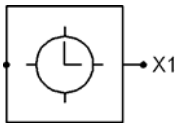
Unit	値
TLOOP	プログラムループ

例: Measure Period





Get Time µs



用途

- OS.LoopCnt 時間をミリ秒(µs)で出力します
- ロジックの実行時間を計測します
- 分解能はハードウェアによって決まります

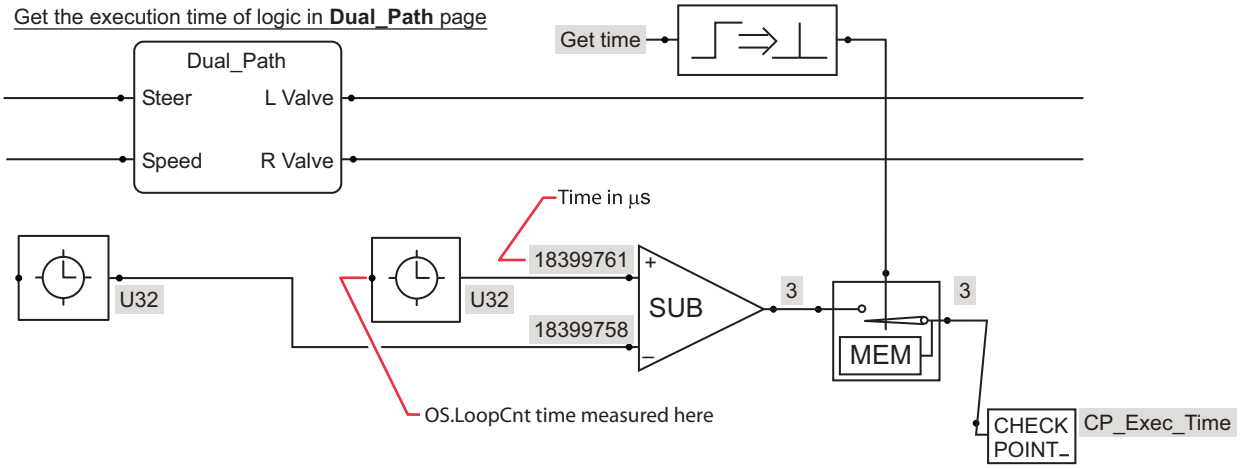
機能

- X1 = コンポーネントの実行時間（ミリ秒単位）

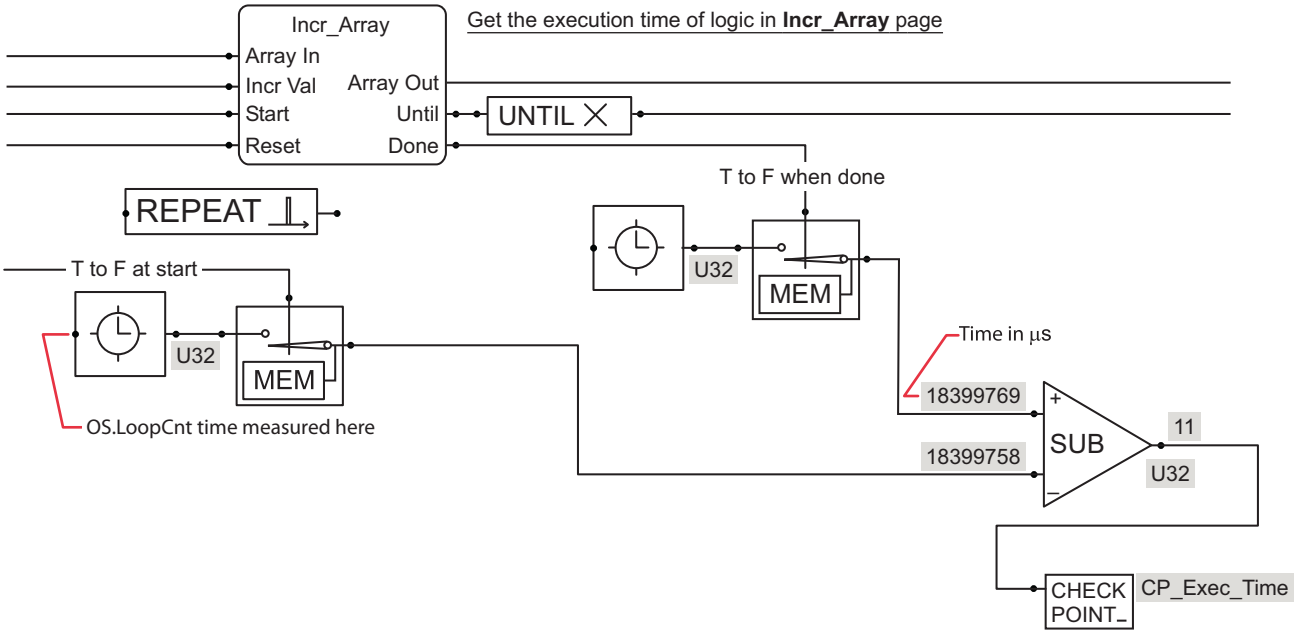
有効接続

ピン	データ型	ピン	データ型
	—	X1	U32

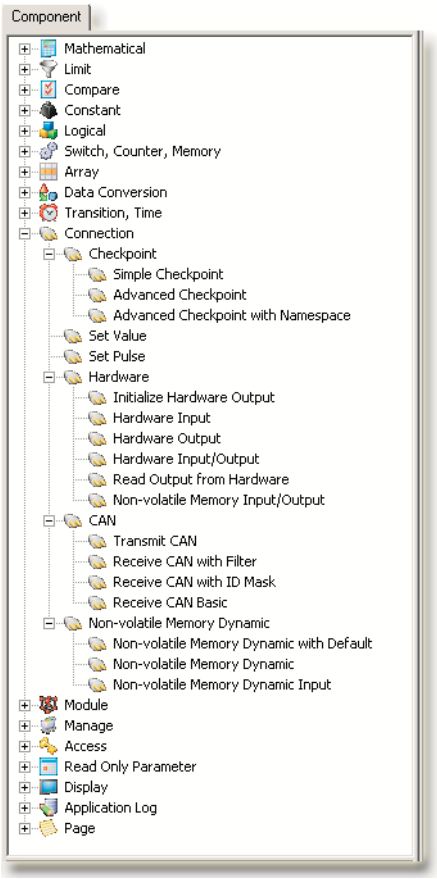
例 1: Get Time µs



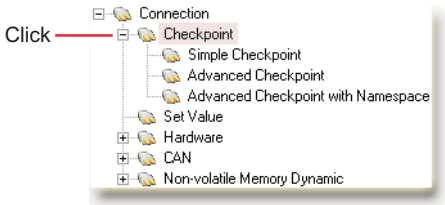
例 2: Get Time µs



Connection（接続）



Checkpoint（チェックポイント）



Simple Checkpoint（単純チェックポイント）

用途

- PLUS+1 サービスツールプログラムでチェックできるチェックポイント信号を生成します。
- チェックしたい信号を A1 に送ります。?入力が信号名となります。
- Query/Change ツールでアンダースコア (_) をクリックして、0-9 の PLUS+1 サービスツールプログラムのアクセスレベルを値に設定します（0 が最も高いアクセスレベルです）。

チェックポイントと不揮発性動的メモリコンポーネントに使用される ? 名は異なっていなければなりません。名が同一であると、MULTI DEFINED SYMBOL コンパイルエラーとなります。

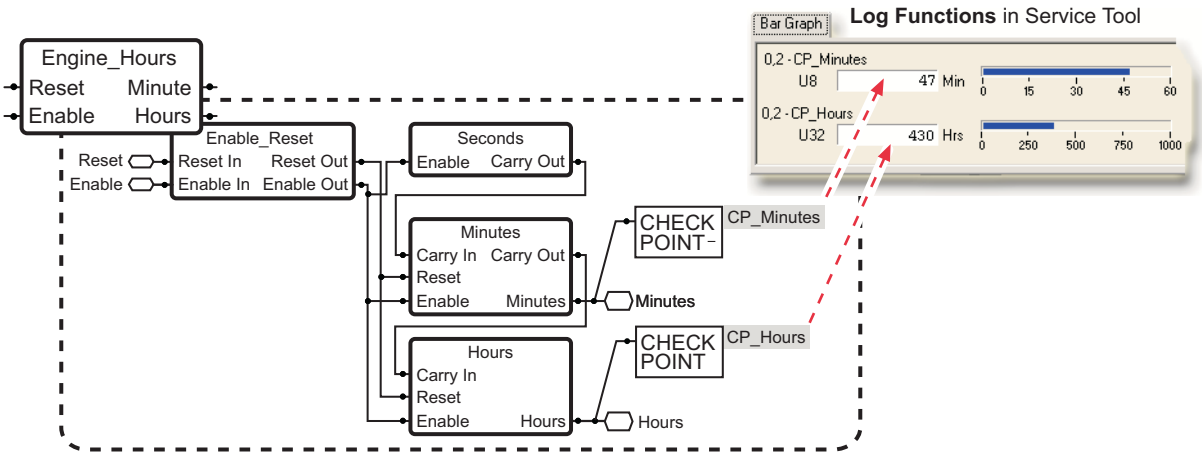
機能

- ? = A1 信号のユーザ定義名

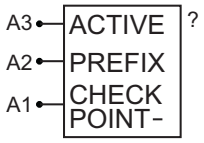
有効接続

ピン	データ型	ピン	データ型
A1	ARRAY, BOOL, INT	—	—

例: Simple Checkpoint



Advanced Checkpoint（アドバンス・チェックポイント）



用途

- PLUS+1 サービスツールプログラムでチェックできるチェックポイント信号を生成します。
- チェックしたい信号を A1 に送ります。? に設定した名前が信号名となります。
- A2 PREFIX を ? 信号名に追加します。

  ページをコピーする場合、A2 PREFIX を変更するだけで結構です。コピーされたページを使用するために、? 名を変更する必要はありません。
- Query/Change ツールでアンダースコア (_) をクリックして、0-9 の PLUS+1 サービスツールプログラムのアクセスレベルを値に設定します。（0 が最も高いアクセスレベルです。）

チェックポイントと不揮発性動的メモリコンポーネントに使用される ? 名は異なっていなければなりません。名が同一であると、MULTI DEFINED SYMBOL コンパイルエラーとなります。

機能

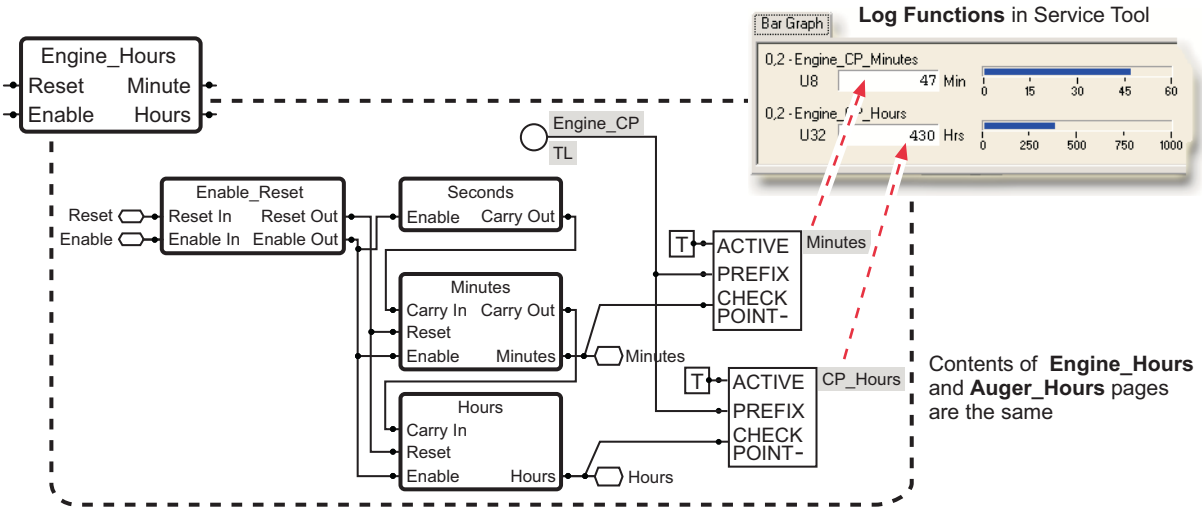
- ? = A1 信号のユーザ定義名
- ? の信号は、
  - A3 = T の場合、PLUS+1 サービスツールプログラムで使用することができます

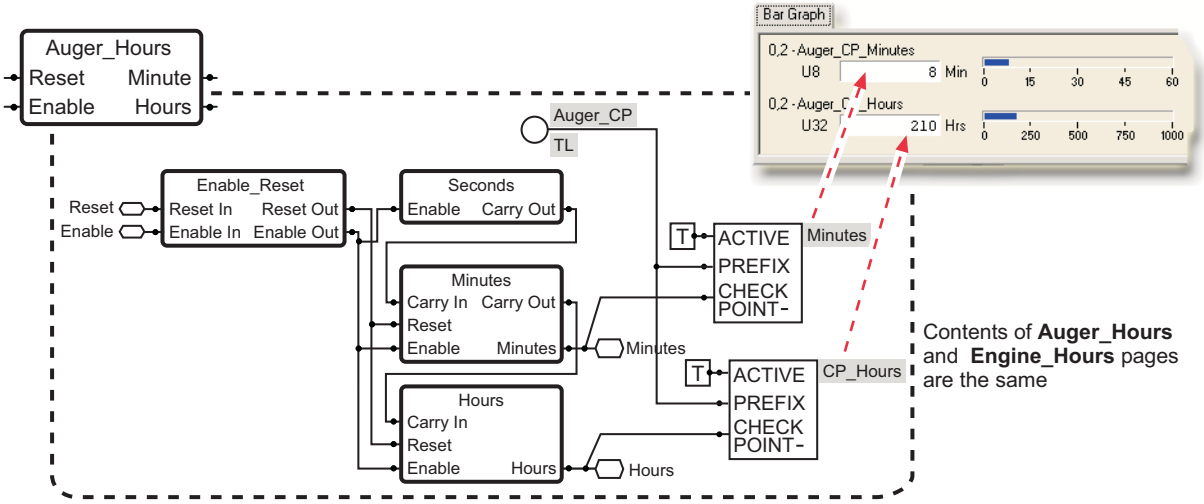
    PLUS+1 サービスツールプログラムでは、?信号名が A2 PREFIX + アンダースコア + ?信号名となります
  - A3 = F の場合、使用することができません

有効接続

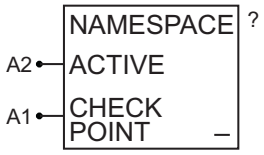
ピン	データ型	ピン	データ型
A1	ARRAY, BOOL, INT	—	—
A2	TL	—	—
A3	BOOL	—	—

例: Advanced Checkpoint





Advanced Checkpoint with Namespace（ネームスペース付きアドバンスチェックポイント）



用途

- PLUS+1 サービスツールプログラムでチェックできるチェックポイント信号を生成します。
- チェックしたい信号を A1 に送ります。? に設定した名前が信号名となります。
- ページのネームスペースを?信号名の前に付加します。

  ページをコピーする場合、ページのネームスペースを変更するだけで結構です。コピーされたページを使用するために、?名を変更する必要はありません。
- Query/Change ツールでアンダースコア (_) をクリックして、0–9 の PLUS+1 サービスツールプログラムのアクセスレベルを値に設定します（0 が最も高いアクセスレベルです）。

チェックポイントと不揮発性動的メモリコンポーネントに使用される ? 名は異なっていなければなりません。名が同一であると、MULTI DEFINED SYMBOL コンパイルエラーとなります。

機能

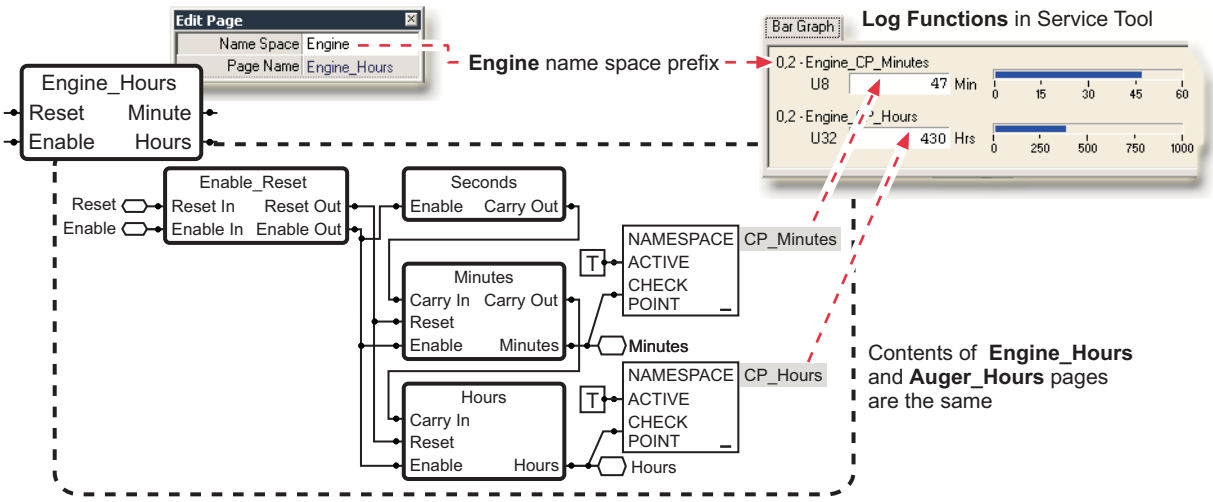
- ? = A1 信号のユーザ定義名
- 以下の場合には
  - A2 = T の場合、PLUS+1 サービスツールプログラムで?信号を使用することができます

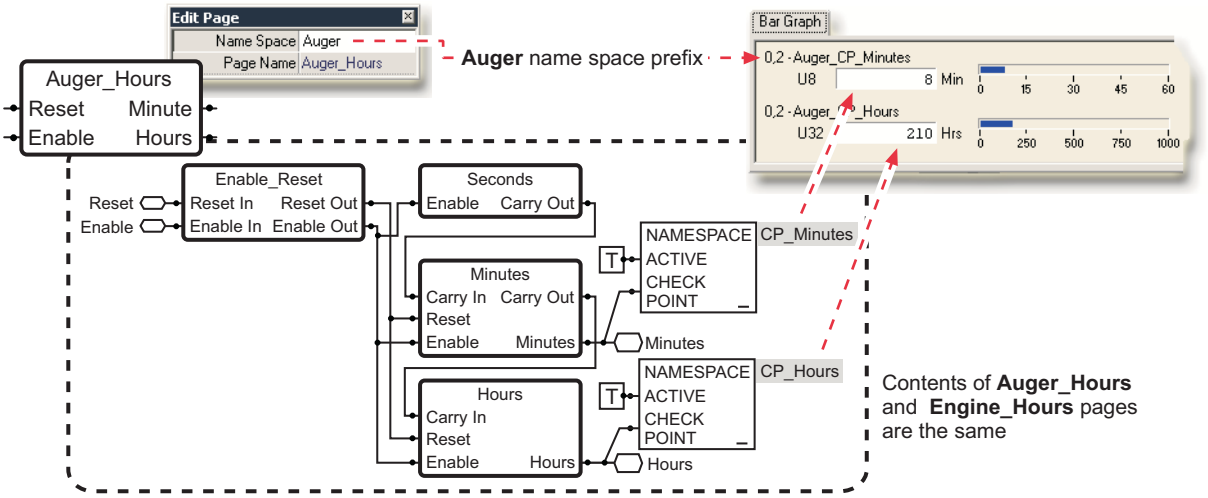
    PLUS+1 サービスツールプログラムでは、?信号名がページのネームスペース + アンダースコア + ?信号名となります
  - A2 = F の場合、?信号を使用することができません

有効接続

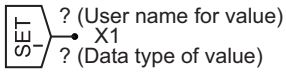
ピン	データ型	ピン	データ型
A1	ARRAY, BOOL, INT	—	—
A2	BOOL	—	—

例: Advanced Checkpoint with Namespace





Set Value（値の設定）



用途

- 値を試す場合。
- メモリコンポーネントを用いて値を読み書きする必要なく、PLUS+1 サービスツールプログラムからコントローラに値を直接入力します。
- コントローラをオフにすると値がすべて 0 に戻ります。
- Query/Change ツールでアンダースコア (_) をクリックして、0–9 の PLUS+1 サービスツールプログラムのアクセスレベルを値に設定します。（0 が最も高いアクセスレベルです。）

機能

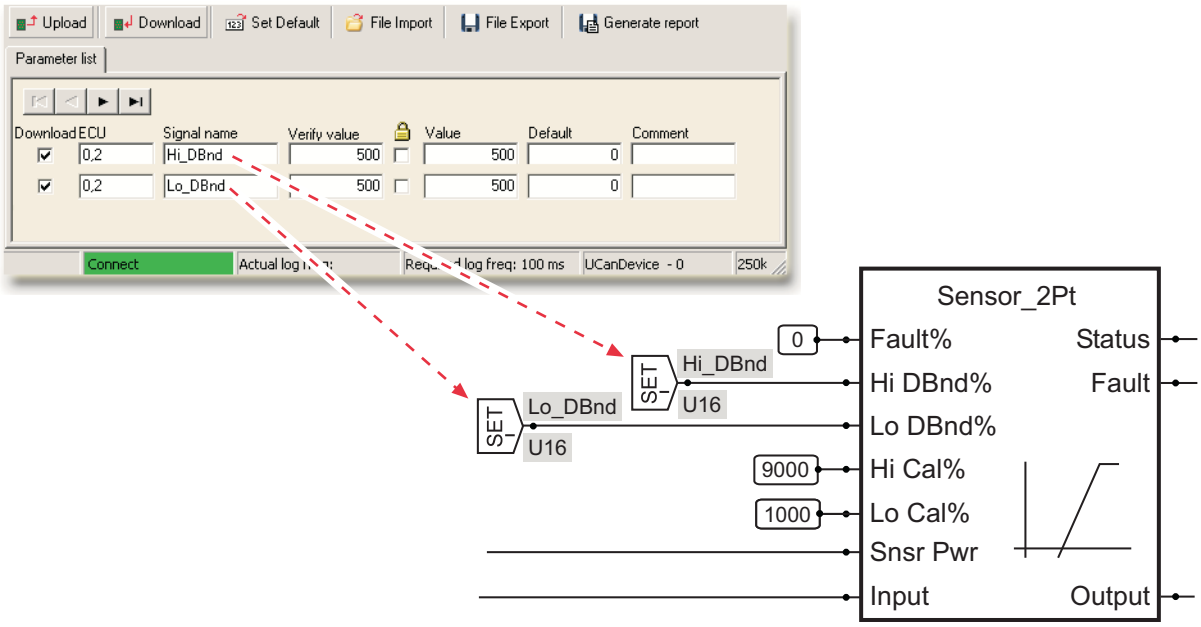
- ? = 値のユーザ名。この名前を用いて、PLUS+1 サービスツールプログラムの値にアクセスします。

有効接続

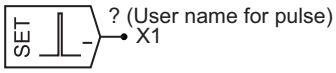
ピン	データ型	ピン	データ型
—	—	X1	ALL

例: Set Value

Parameter Functions in Service Tool



Set Pulse



用途

- PLUS+1 サービスツールプログラムから動作させたとき、1 プログラムループの T/F パルスを与えます
- キャリブレーション値をメモリコンポーネントに書き込む動作を容易に行えます.
- Query/Change ツールでアンダースコアをクリックし、0～9 の PLUS+1 サービスツールプログラムアクセスレベルを設定することができます。(0 が最もアクセスレベルが高い)

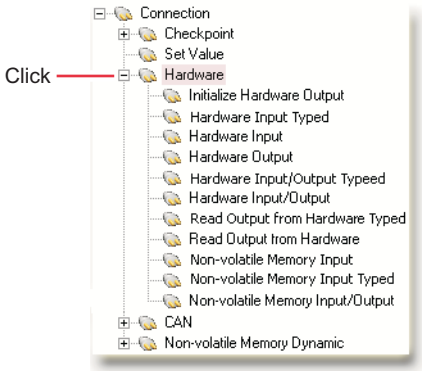
機能

- ? はパルスのユーザ名で、PLUS+1 サービスツールプログラム内のパルスプッシュボタンの名前とリンクしています。

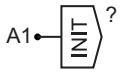
有効接続

ピン	データ型	ピン	データ型
—	—	X1	BOOL

Hardware（ハードウェア）



Initialize Hardware Output（ハードウェア出力初期化）



用途

- 最初のプログラムループの前にカーネルに値を書き込みます。

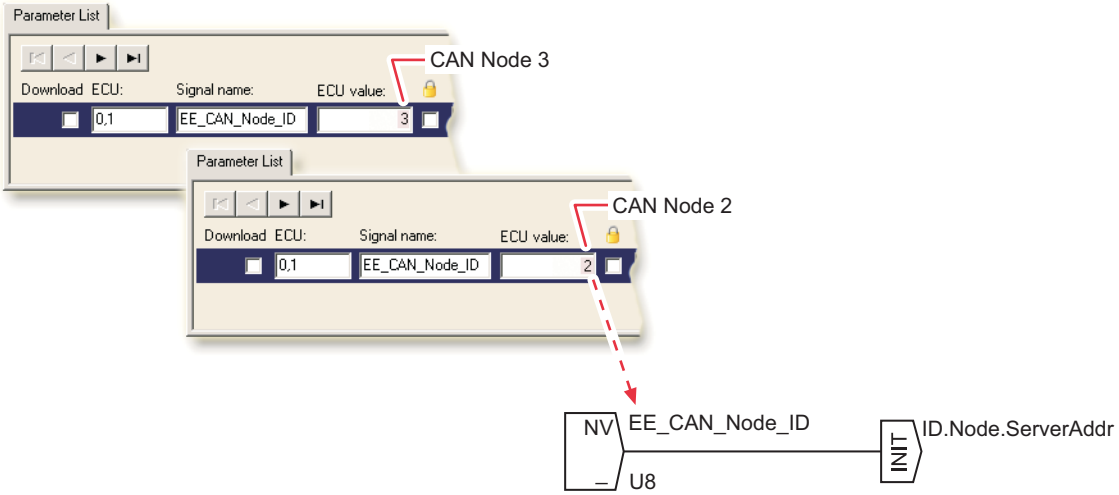
機能

- ? = ハードウェア信号名（完全に一致している必要があります）。信号のリストについてはアプリケーションインタフェース（API）仕様を参照してください
- 定数コンポーネントまたは不揮発性メモリ（EE）コンポーネントからの入力のみ受け付けます
- EE コンポーネントからの入力値を変更した場合、コントローラの電源を再投入した後でのみ有効になります。

有効接続

ピン	データ型	ピン	データ型
A1	ALL	X1	—

例: Initialize Hardware Output

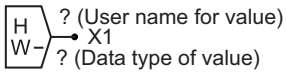


この例では不揮発性メモリコンポーネントを **Initialize Hardware Output** コンポーネントに組み合わせています。

この組み合わせにより、ユーザはサービスツールプログラムを使って同じアプリケーションで **CAN Node 3** と **CAN Node 3** を切替えることが可能になります。ノードスイッチはアプリケーションの再コンパイルやダウンロードなしで実現することができます。

コントローラの電源を再投入することでノードスイッチが有効になります。

Hardware Input Typed（ハードウェア入力、型付き）



用途

- カーネルからアプリケーションに値を読み込みます。

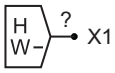
機能

- ? = ハードウェア信号名（完全に一致している必要があります）。信号のリストについてはアプリケーションインタフェース（API）仕様を参照してください

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

Hardware Input（ハードウェア入力）



用途

- カーネルからアプリケーションに値を読み込みます。

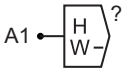
機能

- オートタイプされます
- ? = ハードウェア信号名（完全に一致している必要があります）。信号のリストはアプリケーションインタフェース（API）仕様を参照してください

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

Hardware Output（ハードウェア出力）



用途

- プログラムループの終わり毎にアプリケーションからカーネルに値を書き込みます。
- 出力を1回だけ設定するためには、最初のプログラムループの開始前に、**Initialize Hardware Output**コンポーネント(*271* ページ、*Initialize Hardware Output*).を使用します。

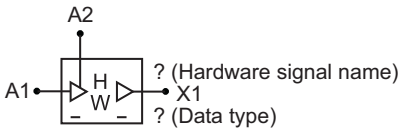
機能

- ? = ハードウェア信号名（完全に一致している必要があります）。信号のリストについてはアプリケーションインタフェース（API）仕様参照

有効接続

ピン	データ型	ピン	データ型
A1	ALL	—	—

Hardware Input/Output Typed（ハードウェア入出力、型付き）



PLUS+1 GUIDE 2.1（およびそれ以降）用に開発されたテンプレートを用いて新しいアプリケーションを作成する場合は、***281***ページに記載されている新しい***CAN***コンポーネントを使用してください。本コンポーネントを使用するアプリケーションを更新する場合（通常はCAN通信用）には、***280***ページの「***Using Old CAN and Memory Components***」を参照してください。

用途

- カーネルからの読み取りとカーネルへの書き込みを切り替えます。

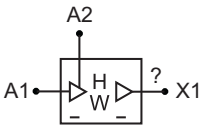
機能

- ? = ハードウェア信号名（完全に一致している必要があります）。信号のリストについてはアプリケーションインタフェース（API）仕様を参照のこと
- A2 = F（A1 が X1 に接続されていない）であれば X1 = カーネルから読み取り
- A2 = T（A1 = X1）であれば X1 = カーネルへ書き込み

有効接続

ピン	データ型	ピン	データ型
A1	ALL	X1	ALL
A2	BOOL		

Hardware Input/Output（ハードウェア入出力）



PLUS+1 GUIDE 2.1（およびそれ以降）用に開発されたテンプレートを用いて新しいアプリケーションを作成する場合は、***281***ページに記載されている新しい***CAN***コンポーネントを使用してください。（通常はCAN通信のために）本コンポーネントを使用するアプリケーションを更新する場合には、***280***ページの「***Using Old CAN and Memory Components***」を参照してください。

用途

- カーネルからの読み取りとカーネルへの書き込みを切り替えます。

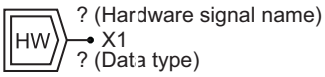
機能

- オートタイプされます
- ? = ハードウェア信号名（完全に一致している必要があります）。信号のリストについてはアプリケーションインタフェース（API）仕様を参照のこと
- A2 =F (A1 が X1 に接続されていない)であれば X1 = BIOS 入力
- A2 =T (A1 = X1)であれば X1 = BIOS 出力

有効接続

ピン	データ型	ピン	データ型
A1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG	X1	BOOL, INT, TIME, TEXT, FILE, FONT, PORT, COL, IMG
A2	BOOL		

Read Output from Hardware Typed（ハードウェアからの出力読み取り、型付き）



用途

- カーネル出力信号を読み取ります

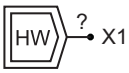
機能

- ? = ハードウェア信号名（完全に一致している必要があります）。信号のリストについてはアプリケーションインタフェース（API）仕様書を参照してください。

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

Read Output from Hardware（ハードウェアからの出力読み取り）



用途

- カーネル出力信号を読取ります。
- X1 出力はオートタイプされます。

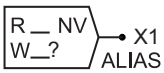
機能

- ? = ハードウェア信号名（完全に一致している必要があります）。信号のリストについてはアプリケーションインタフェース（API）仕様書を参照してください

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

Non-Volatile Memory Input Typed（不揮発性メモリ入力、型付き）



本コンポーネントを使用している古いアプリケーションを更新する場合には、***280***ページの「***Using Old CAN and Memory Components***」を参照してください。
PLUS+1 GUIDE 2.1（およびそれ以降）用に開発されたテンプレートを用いて新しいアプリケーションを作成する場合は、***292***ページに記載されている新しい***Non-Volatile Memory Dynamic***コンポーネントを使用してください。

用途

- 不揮発性メモリへの入力用接続ポイント。

機能

- ? = ハードウェア信号名（完全に一致している必要があります）

  ハードウェア信号名をタイプインする必要があります

- ALIAS を PLUS+1 サービスツールプログラムに表示させたい名前に置き換えます

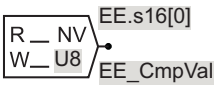
  ALIAS を変更しない場合、PLUS+1 サービスツールプログラムではハードウェア名が使用されます

有効接続

ピン	データ型	ピン	データ型
—	—	X1	INT

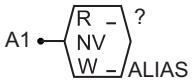
例: Non-Volatile Memory Input Typed

X1 ouputs the S16 value stored in memory locationEE.s16[0].



In the Service Tool, select the EE_CmpVal signal to view and change the value stored in memory location EE.s16[0].

Non-Volatile Memory Input/Output（不揮発性メモリ入力/出力）



本コンポーネントを使用している古いアプリケーションを更新する場合には、***280***ページの「***Using Old CAN and Memory Components***」を参照してください。
PLUS+1 GUIDE 2.1（およびそれ以降）用に開発されたテンプレートを用いて新しいアプリケーションを作成する場合は、***292***ページに記載されている新しい***Non-Volatile Memory Dynamic***コンポーネントを使用してください。

用途

- 不揮発性メモリへの BIOS 出力用接続ポイント

機能

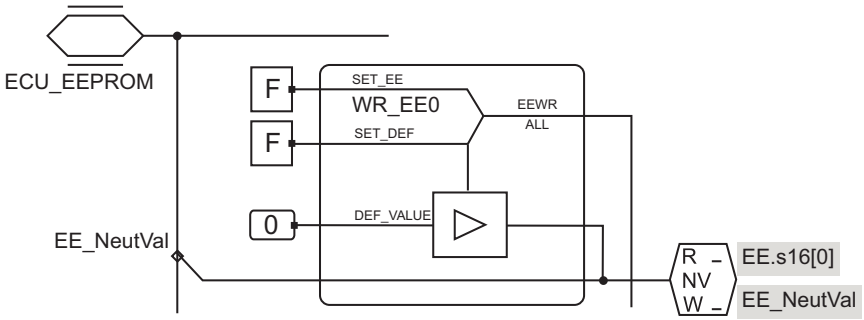
- A1 = BIOS ハードウェア信号名（完全に一致している必要があります）
- ? =データのソフトウェアロケーション
- ALIAS を PLUS+1 サービスツールプログラムに表示させたい名前に置き換えます

  ALIAS を変更しない場合は、PLUS+1 サービスツールプログラムでは BIOS(EE.s ***nn***[***n***]) ハードウェア名が使用されます

有効接続

ピン	データ型	ピン	データ型
A1	ALL	—	—

例: Non-Volatile Memory Input/Output



In the Service Tool, select the EE_NeutVal signal to view and change the value stored in memory location EE.s16[0].

Using Old CAN and Memory Components（古いCANとメモリコンポーネントの使用）

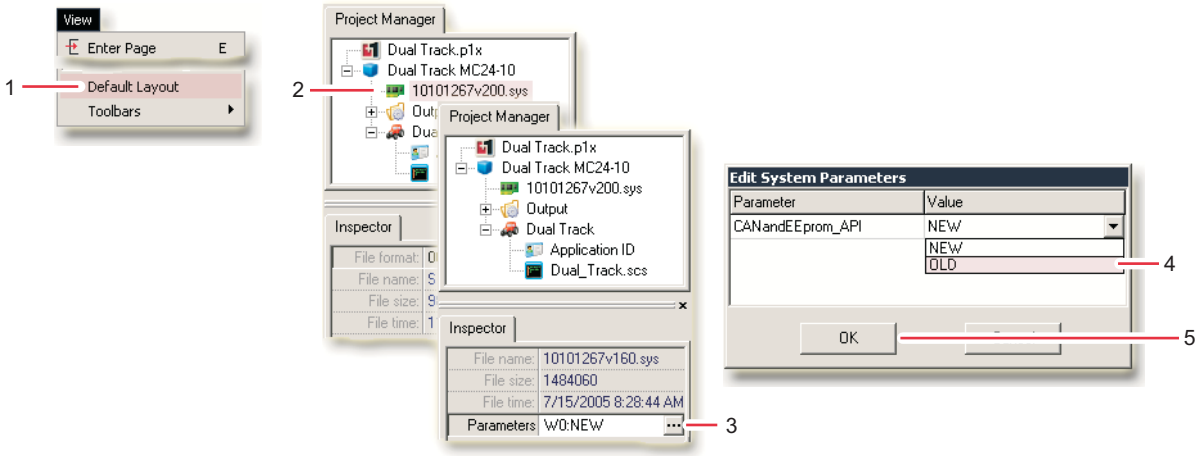
この表には、新旧の CAN とメモリ（EEprom）コンポーネントが記載されています。PLUS+1 GUIDE 2.1 ソフトウェアでは、**Old** コンポーネントを **New** コンポーネントに置き換えています。

新旧の CAN とメモリコンポーネント

Old	New
Non-Volatile Memory Input （不揮発性メモリ入力）	Transmit CAN（CAN 送信）
Non-Volatile Memory Input Typed （不揮発性メモリ入力、タイプ付き）	Receive CAN with Filter（フィルタ付き CAN 受信）
Hardware Input/Output Typed （ハードウェア入力/出力、タイプ付き）	Receive CAN with ID Mask（ID マスク付き CAN 受信）
Hardware Input/Output Typed （ハードウェア入力/出力、タイプ付き）	Receive CAN Basic（CAN ベーシック受信）
	Non-Volatile Memory Dynamic with Default （デフォルト付き不揮発性動的メモリ）
	Non-Volatile Memory Dynamic （不揮発性動的メモリ）
	Non-Volatile Memory Input（不揮発性メモリ入力）

- PLUS+1 GUIDE 2.1（およびそれ以降）ソフトウェア用に開発されたテンプレートを用いて作成されたアプリケーションは、**New** コンポーネントを使用しなければなりません。
- PLUS+1 GUIDE 2.1（およびそれ以降）ソフトウェアがこの表に記載されている**OLD**コンポーネントを用いてアプリケーションをコンパイルできるようにするためには、システムファイルパラメータを**NEW**から**OLD**に変更しなければなりません。*281*ページの「*Change from New to Old CAN and Memory Components*」を参照してください。
- システムファイルパラメータが **OLD** に設定された PLUS+1 GUIDE 2.1（およびそれ以降）ソフトウェアは、この表に記載されている **New** コンポーネントを用いてアプリケーションをコンパイルすることはできません。

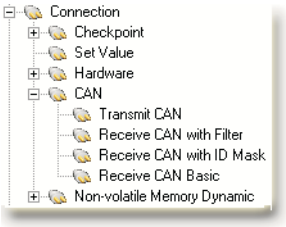
Change from New to Old CAN and Memory Components（新しいCANとメモリコンポーネントから古いCANとメモリコンポーネントへの変更）



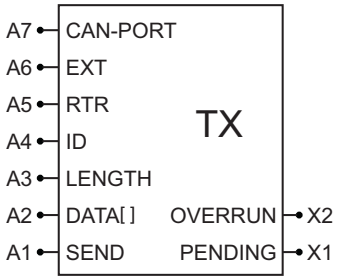
システムファイルパラメータを **NEW** から **OLD** に変更するには

1. **View** メニューで、**Default Layout** をクリックして、**Project Manager** タブと **Inspector** タブを表示します。
2. **Project Manager** タブで、**System file** をクリックします。
3. **Inspector** タブで、クリックして **Edit System Parameters**（システムパラメーター編集）ウインドウを表示します。
4. **Edit System Parameters** ウインドウで、**OLD** をクリックします。
5. **Edit System Parameters** ウインドウで、**OK** をクリックします。

CAN


Connection
Checkpoint
Set Value
Hardware
CAN
Transmit CAN
Receive CAN with Filter
Receive CAN with ID Mask
Receive CAN Basic
Non-volatile Memory Dynamic

Transmit CAN（CAN送信）



PLUS+1 GUIDE 2.1 ソフトウェアのリリース前に作成されたアプリケーションを更新する場合は、本コンポーネントを使用する前に*280*ページの「*Using Old CAN and Memory Components*」を見てください。

このコンポーネントをモジュール内で使用するとき、A1（**SEND**）＝T ならばモジュールを無効化しても送信を継続します。これを防ぐために、A1=F に設定するまでモジュールを有効化しておくようなロジックを使用してください。

用途

- CAN メッセージを送信します。

機能

- A1：A2 のメッセージを送信するときに T にセット
- A2 = メッセージデータフィールドの配列、データバイト 0 から読み取り
- A3 = データ長コード、データフィールド内のバイト数 (0–8)
- A4 = メッセージ ID (識別子フィールド)
- A5：
  - A5 = T の場合、RTR（Remote Transmission Request）フラグをセット（J1939 ではサポートされていません）
  - A5 = F の場合、RTR フラグを解除
- A6：
  - A6 = F の場合、標準 11 ビットメッセージ ID フィールド (CAN 2.0A) となります
  - A6 = T の場合、拡張 29 ビットメッセージ ID フィールド (CAN 2.0B) となります
- A7：ハードウェアファイルにより定義されている CAN ポートに接続（ハードウェアの API ドキュメントを参照）

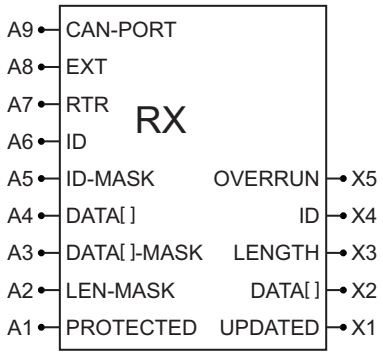
- X1:
  - X1 = T の場合、メッセージが保留状態です
  - X1 = F の場合、メッセージが保留状態ではありません
- X2:
  - X2 = F の場合、オーバランはありません
  - X2 = T の場合、送信されていないメッセージが保留状態（X1=T）で新しいメッセージ (A1 = T)となっています。送信されていないメッセージは失われます

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	ARRAY[]U8	X2	BOOL
A3	INT		
A4	UINT		
A5	BOOL		
A6	BOOL		
A7	PORT		

本コンポーネントを用いた***291***ページの *例:* ***Receive CAN Basic***を参照してください。

Receive CAN with Filter（フィルタ付きCAN受信）



PLUS+1 GUIDE 2.1 ソフトウェアのリリース前に作成されたアプリケーションを更新する場合は、本コンポーネントを使用する前に***280***ページの「***Using Old CAN and Memory Components***」を参照してください。

用途

- CAN メッセージを受信します。
- ID 入力にマスキングを適用します。
- DATA 入力にマスキングを適用します。

機能

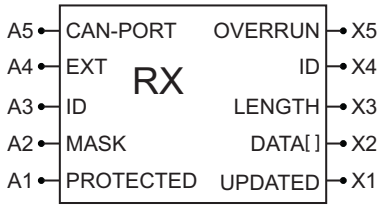
- A1：
  - A1 = T の場合、プログラムループで受信した最初のメッセージを保存します。最初のメッセージに後続のメッセージを上書きしません
  - A1 = F の場合、プログラムループで受信した最後のメッセージを保存します。最後のメッセージを先行するメッセージに上書きします
- A2 = 使用する DATA[ ]-MASK 内のバイト数
- A3 = A4 データに適用されるマスク
- A4 = データ
- A5 = メッセージ ID に適用されるマスク (識別子フィールド)
- A6 = メッセージ ID (識別子フィールド)

- A7:
  - A7 = T の場合、送信要求 (RTR) メッセージのみを受信します
  - A7 = F の場合、RTR メッセージを受信しません
- A8:
  - A8 = F の場合、標準 11 ビットメッセージ ID フィールド (CAN 2.0A) となります
  - A8 = T の場合、拡張 29 ビットメッセージ ID フィールド (CAN 2.0B) となります
- A9: ハードウェアファイルにより定義されている CAN バスのポート信号に接続されます
- X1:
  - X1 = T の場合、最後のプログラムループ以降に受信された新しいメッセージがあります
  - X1 = F の場合、最後のプログラムループ以降に受信された新しいメッセージはありません
- X2 = メッセージデータフィールドの配列、データバイト 0 から読み取り
- X3 = データ長コード、データフィールド内のバイト数 (0–8)
- X4 = メッセージ ID
- X5:
  - X5 = T の場合、プログラムループ中に複数のメッセージが受信されています
  - X5 = F の場合、オーバランはありません

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	UINT	X2	ARRAY [8] U8
A3	ARRAY [ ] U8	X3	U8
A4	ARRAY [ ] U8	X4	U32
A5	UINT	X5	BOOL
A6	UINT		
A7	BOOL		
A8	BOOL		
A9	PORT		

Receive CAN with ID Mask（IDマスク付きCAN受信）



PLUS+1 GUIDE 2.1 ソフトウェアのリリース前に作成されたアプリケーションを更新する場合は、本コンポーネントを使用する前に***280***ページの「***Using Old CAN and Memory Components***」を見てください。

用途

- CAN メッセージを受信します。
- メッセージ ID 入力にマスキングを適用します。

機能

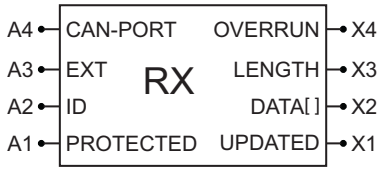
- A1：
  - A1 = T の場合、プログラムループで受信した最初のメッセージを保存します。最初のメッセージに後続のメッセージを上書きしません
  - A1 = F の場合、プログラムループで受信した最後のメッセージを保存します。最後のメッセージを先行するメッセージに上書きします
- A2 = メッセージ ID（識別子フィールド）に適用されるマスクを定義します。マスクビットが 0 の場合には、「無関係」となります
- A3 = メッセージ ID（識別子フィールド）
- A4：
  - A4 = F の場合、標準 11 ビットメッセージ ID フィールド（CAN 2.0A）となります
  - A4 = T の場合、拡張 29 ビットメッセージ ID フィールド（CAN 2.0B）となります
- A5：ハードウェアファイルにより定義されている CAN バスのポート信号に接続（ハードウェアの API ドキュメントを参照）
- X1：
  - X1 = T の場合、最後のプログラムループ以降に受信された新しいメッセージがあります
  - X1 = F の場合、最後のプログラムループ以降に受信された新しいメッセージはありません

- X2 = メッセージデータフィールドの配列、データバイト 0 から読み取り
- X3 = データ長コード、データフィールド内のバイト数 (0–8)
- X4 = 実際のメッセージ ID
- X5:
  - X5 = T の場合、プログラムループ中に複数のメッセージが受信されています
  - X5 = F の場合、オーバランはありません

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	UINT	X2	ARRAY [8] U8
A3	UINT	X3	U8
A4	BOOL	X4	U32
A5	PORT	X5	BOOL

Receive CAN Basic（CAN受信ベーシック）



PLUS+1 GUIDE 2.1 ソフトウェアのリリース前に作成されたアプリケーションを更新する場合は、本コンポーネントを使用する前に***280***ページの「***Using Old CAN and Memory Components***」を見てください。

用途

- CAN メッセージを受信します。

機能

- A1：
  - A1 = T の場合、プログラムループで受信した最初のメッセージを保存します。最初のメッセージに後続のメッセージを上書きしません
  - A1 = F の場合、プログラムループで受信した最後のメッセージを保存します。最後のメッセージを先行するメッセージに上書きします
- A2 = メッセージ ID（識別子フィールド）
- A3：
  - A3 = F の場合、標準 11 ビットメッセージ ID フィールド (CAN 2.0A) となります
  - A3 = T の場合、拡張 29 ビットメッセージ ID フィールド (CAN 2.0B) となります
- A4：ハードウェアファイルにより定義されている CAN バスのポート信号に接続
  - MC024 マイクロコントローラの場合、CAN_0 バス
  - MC050 マイクロコントローラの場合、CAN_0 バスまたは CAN_1 バス
- X1：
  - X1 = T の場合、最後のプログラムループ以降に受信された新しいメッセージがあります
  - X1 = F の場合、最後のプログラムループ以降に受信された新しいメッセージはありません
- X2 = メッセージデータフィールドの配列、データバイト 0 から読み取り
- X3 = データ長コード、データフィールド内のバイト数 (0–8)
- X4：
  - X4 = T の場合、プログラムループ中に複数のメッセージが受信されています

- X4 = F の場合、オーバランはありません

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	BOOL
A2	UINT	X2	ARRAY[8]U8
A3	BOOL	X3	U8
A4	PORT	X4	BOOL

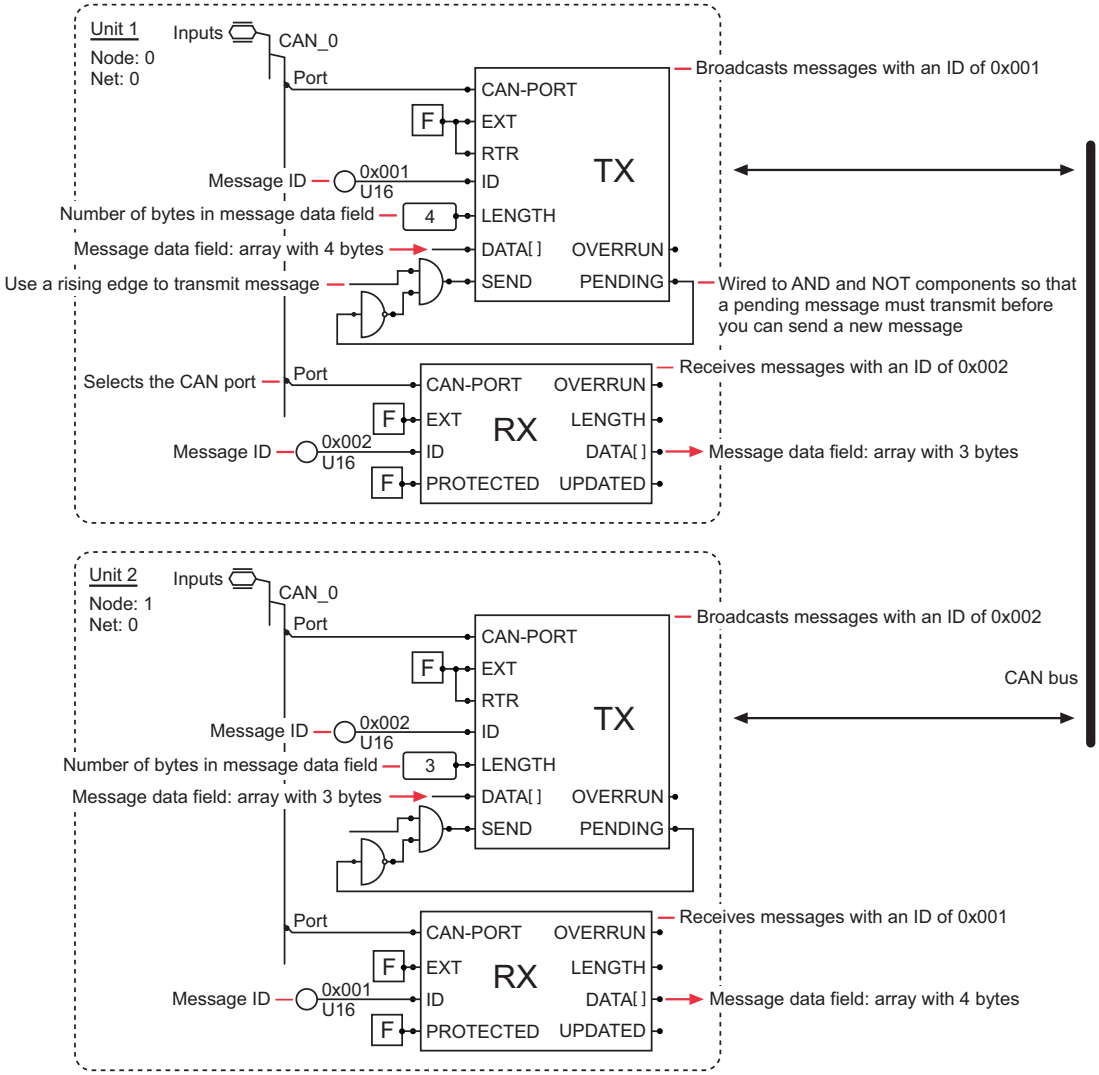
例: Receive CAN Basic

ユニット 1 では

- **Transmit CAN** コンポーネント (TX) が ID=0x001 のメッセージを送信します。
- **Receive CAN Basic** コンポーネント(RX) は ID=0x002 のメッセージを受信します。

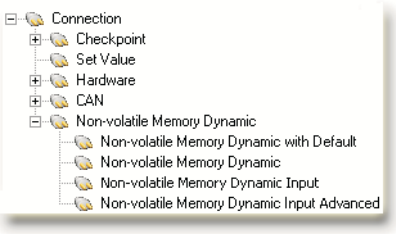
ユニット 2 では

- **Transmit CAN** コンポーネント (TX) が ID=0x002 のメッセージを送信します。
- **Receive CAN Basic** コンポーネント(RX) は ID=0x001 のメッセージを受信します。

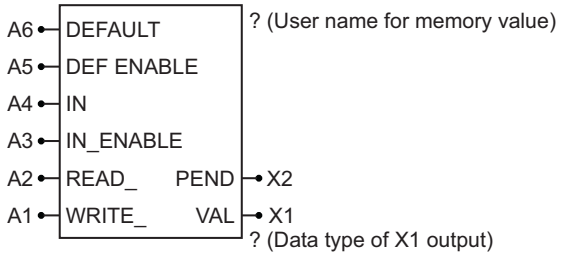


- **Set Array Element**コンポーネントを用いて、動的ランタイム値を含むメッセージデータフィールドを構成する方法に関しては、***221***ページの *例 3: Set Array Element*を参照してください。
- **Get Array Element**コンポーネントを用いてメッセージデータフィールド内の個々の要素を出力する方法に関しては、***219***ページの *例 2: Get Array Element*を参照してください。

Non-Volatile Memory Dynamic（不揮発性動的メモリ）



Non-Volatile Memory Dynamic with Default（デフォルト付き不揮発性動的メモリ）



PLUS+1 GUIDE 2.1 ソフトウェアのリリース前に作成されたアプリケーションを更新する場合は、本コンポーネントを使用する前に***280***ページの「***Using Old CAN and Memory Components***」を見てください。

用途

- 不揮発性メモリ値を読み書きします。
- デフォルト値をメモリに書き込みます。
- ハードウェアの電源がオンになると、メモリ内の値が X1 にコピーされます。

機能

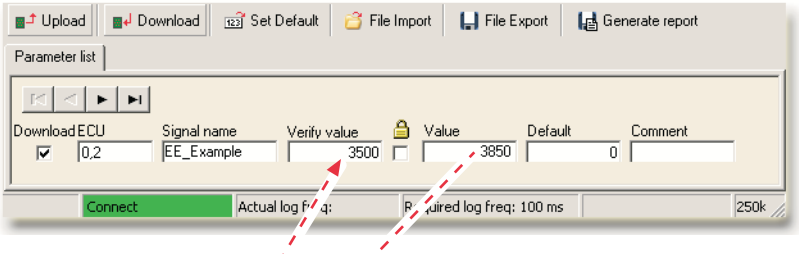
- ? = メモリ値のための名前。PLUS+1 サービスツールプログラムでは、この名前を用いてメモリ値にアクセスします
- A1 = T であれば X1 値がメモリに書き込まれます
- A2 = T であればメモリ内の値が X1 にコピーされます
- A3 = T であれば A4 値が X1 にコピーされます
- A4 = 新しい値
- A5 = T であれば A6 値が X1 にコピーされます
- A6 = デフォルト値
- X1 = メモリから読み取られる値/メモリに書き込まれる値
- X2 = T のときは X1 値のメモリへの書き込みが完了していません

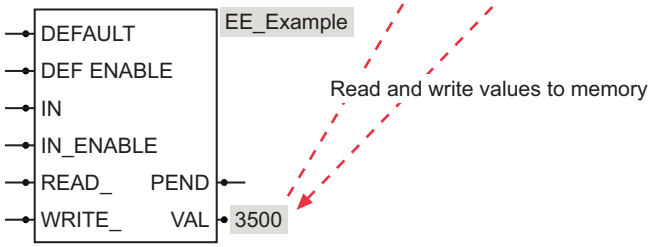
有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	INT, BOOL
A2	BOOL	X2	BOOL
A3	BOOL		
A4	INT, BOOL		
A5	BOOL		
A6	INT, BOOL		

例 1: Non-Volatile Memory Dynamic with Default

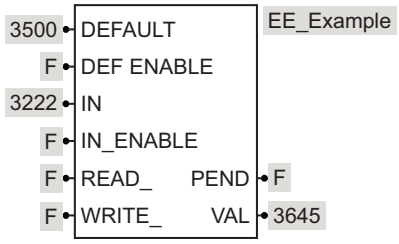
Parameter Functions in Service Tool



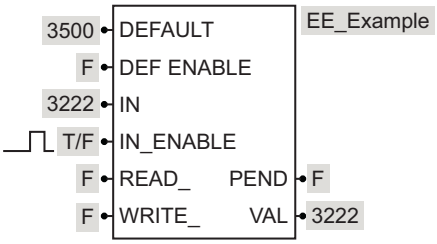


例 2: Non-Volatile Memory Dynamic with Default

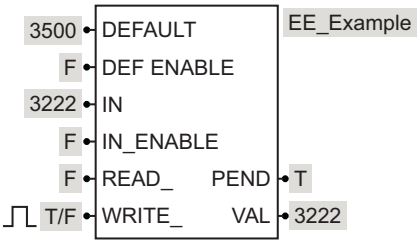
1 Power-up—Value in memory copies to **VAL**



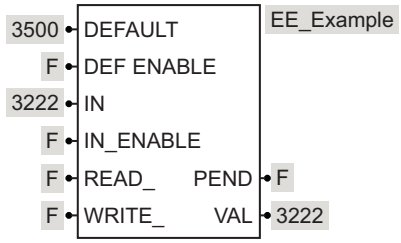
2 **IN** copies to **VAL**



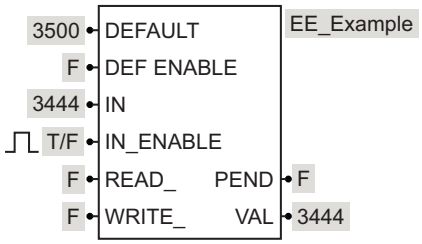
3 **VAL** writes to memory; **PEND** goes **T**



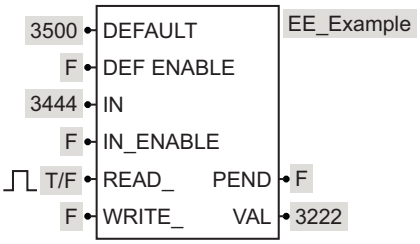
4 Write ends; **PEND** goes **F**



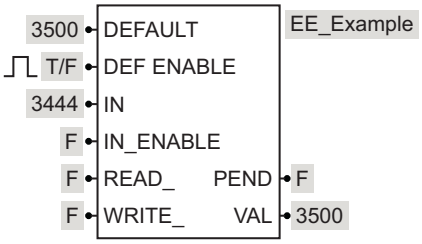
1 **IN** copies to **VAL**



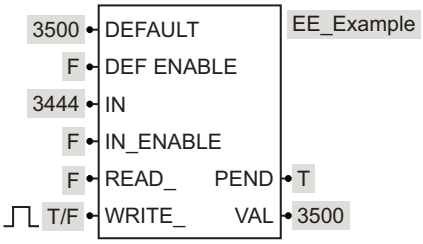
2 Value in memory copies back to **VAL**



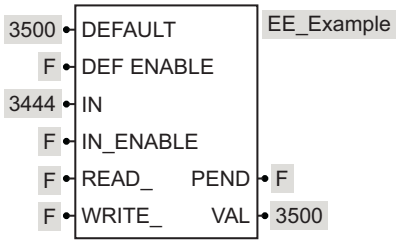
1 **DEFAULT** copies to **VAL**



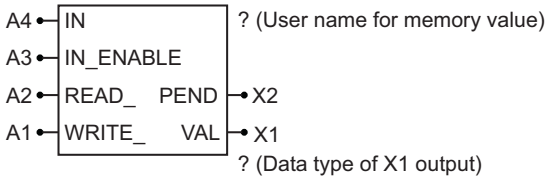
2 **VAL** writes to memory; **PEND** goes **T**



3 Write ends; **PEND** goes **F**



Non-Volatile Memory Dynamic（不揮発性動的メモリ）



PLUS+1 GUIDE 2.1 ソフトウェアのリリース前に作成されたアプリケーションを更新する場合は、本コンポーネントを使用する前に ***280***ページの「***Using Old CAN and Memory Components** 古いCANとメモリコンポーネントの使用」を見てください。

用途

- 不揮発性メモリ値を読み書きします。
- ハードウェアの電源がオンになると、メモリ内の値が X1 にコピーされます。

機能

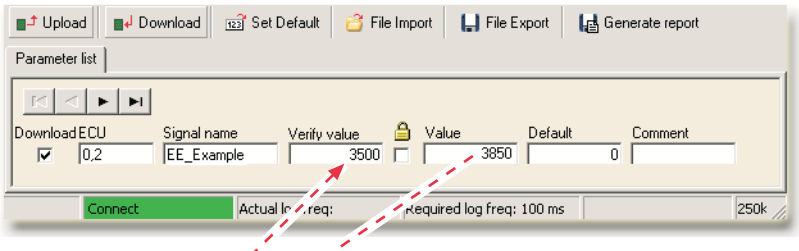
- ? = メモリ値のための名前。PLUS+1 サービスツールプログラムでは、この名前を用いてメモリ値にアクセスします
- A1 = T であれば X1 値がメモリに書き込まれます
- A2 = T であればメモリ内の値が X1 にコピーされます
- A3 = T であれば A4 値が X1 にコピーされます
- A4 = 新しい値
- X1 = メモリから読み取られる値/メモリに書き込まれる値
- X2 = T のときは X1 値のメモリへの書き込みが完了していません

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	INT, BOOL
A2	BOOL	X2	BOOL
A3	BOOL		
A4	INT, BOOL		

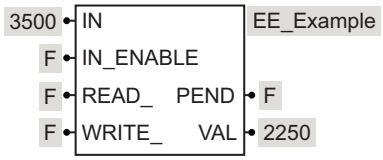
例 1: Non-Volatile Memory Dynamic

Parameter Functions in Service Tool

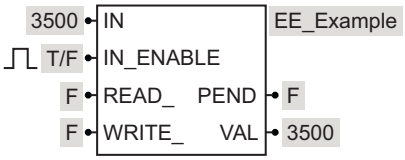


例 2: Non-Volatile Memory Dynamic

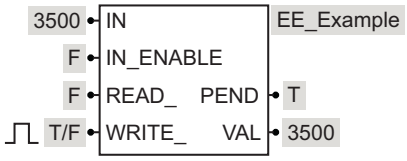
1 Power-up—Value in memory copies to **VAL**



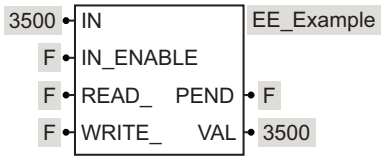
2 **IN** copies to **VAL**



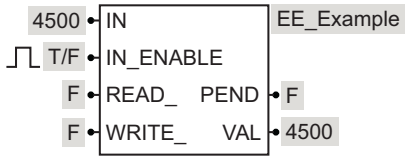
3 **VAL** writes to memory; **PEND** goes **T**



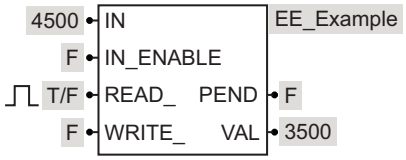
4 Write ends, **PEND** goes **F**



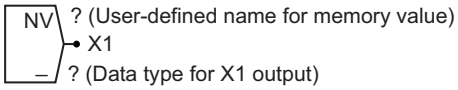
1 **IN** copies to **VAL**



2 Value in memory copies to **VAL**



Non-Volatile Memory Input（不揮発性メモリ入力）



PLUS+1 GUIDE 2.1 ソフトウェアのリリース前に作成されたアプリケーションを更新する場合は、本コンポーネントを使用する前に*280*ページの「*Using Old CAN and Memory Components*」を参照してください。

用途

- PLUS+1 サービスツールプログラムを使用して、不揮発性メモリに直接値を書き込みます。
- アプリケーションからは入力あるいは捕捉できない値を直接設定します
- このメモリ値のサービスツールプログラムからの読み取り/書き込み権限を制限する値を設定します。

機能

- ? = メモリ値のための名前。PLUS+1 サービスツールプログラムでは、この名前を用いてメモリ値にアクセスします。
- _ はオプションで 0～9 の読み取り/書き込みアクセス値を設定します。**Read Access**コンポーネント（*328*ページ参照）と**Write Access**コンポーネント（*329*ページ参照）と共に使用してサービスツールプログラムのこのメモリ値への読み取り/書き込み権限を制限します。
- X1 = メモリの値

Non-Volatile Memory Dynamic Input (with Individual Read/Write Access)
（不揮発性動的メモリ入力（個別読み取り/書き込みアクセス付き））

? (User-defined name for memory value)
X1
? (Data type for X1 output)

PLUS+1 GUIDE 2.1 ソフトウェアのリリース前に作成されたアプリケーションを更新する場合は、本コンポーネントを使用する前に***280***ページの「***Using Old CAN and Memory Components***」を参照してください。

用途

- PLUS+1 サービスツールプログラムを使用して、不揮発性メモリに直接値を書き込みます。
- アプリケーションからは入力あるいは捕捉できない値を直接設定します。
- このメモリ値のサービスツールプログラムからの読み取り/書き込み権限を制限する値を、読み取りと書き込みで別々に設定します。

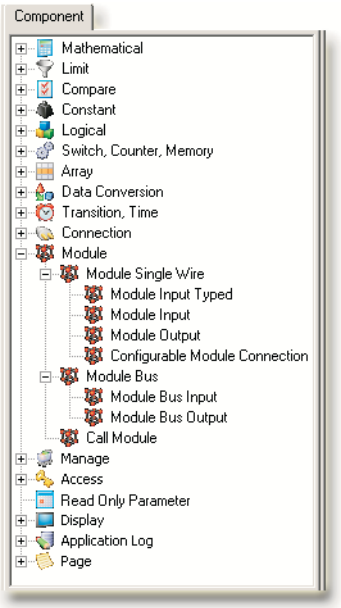
機能

- ? = メモリ値のための名前。PLUS+1 サービスツールプログラムでは、この名前を用いてメモリ値にアクセスします。
- R_ はオプションで 0～9 の読み取りアクセス値を設定します。**Read Access** コンポーネント（***328***ページ参照）と共に使用してサービスツールプログラムのこのメモリ値への読み取り権限を制限します。
- W_ はオプションで 0～9 の書き込みアクセス値を設定します。**Write Access** コンポーネント（***329***ページ参照）と共に使用してサービスツールプログラムのこのメモリ値への書き込み権限を制限します。
- X1 = メモリの値

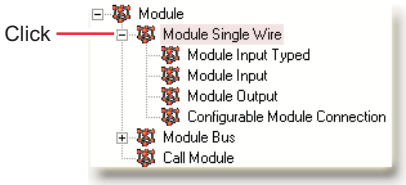
有効接続

ピン	データ型	ピン	データ型
		X1	INT, BOOL

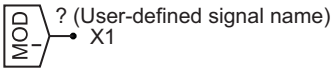
Module（モジュール）



Module Single Wire（モジュール単一ワイヤ）



Module Input（モジュール入力）



用途

- モジュール信号用入力ポイント。
- Query/Change ツールでアンダースコア (_) をクリックして、0-9 の PLUS+1 サービスツールプログラムのアクセスレベルを設定します（0 が最も高いアクセスレベルです）。

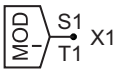
機能

- ? = ユーザ定義信号名
- X1 = 入力信号

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

Module Input Typed（型付きモジュール入力）



用途

- モジュール信号用入力ポイント
- Query/Change ツールでアンダースコア（_）をクリックして、0～9 の PLUS+1 サービスツールプログラムのアクセスレベルを設定します（0 が最も高いアクセスレベルです）。

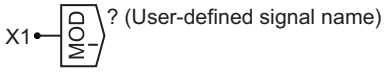
機能

- ? = ユーザ定義信号名
- X1 = 入力信号

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

Module Output（モジュール出力）



用途

- モジュール信号に関する出力を行います。

機能

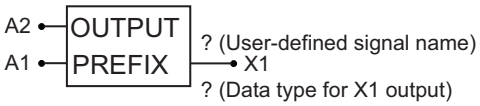
- ? = ユーザ定義信号名
- X1 = 出力信号

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

本コンポーネントの使い方は ***306*** ページの ***Call Module*** を参照してください。

Configurable Module Connection（構成可能モジュール接続）



用途

- 2つのモジュール間の接続を行います。
  - モジュールから信号を出力します。
  - 同じ信号を別のモジュールに入力します。

    Export Block コマンドと **Import Block** コマンドを用いて、接続のエクスポートとインポートを行います。

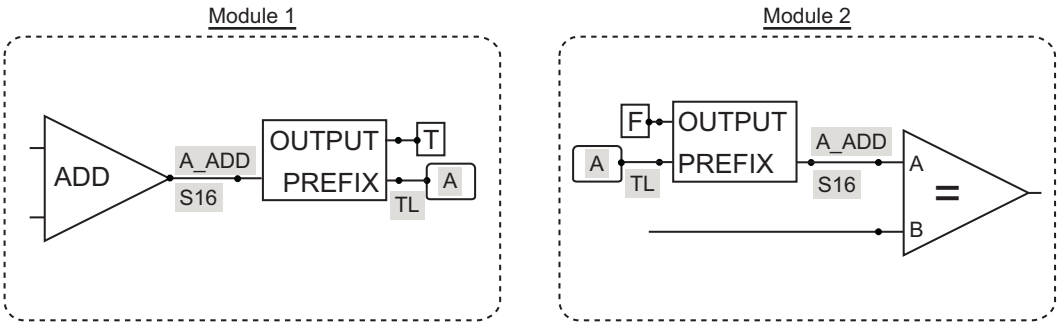
機能

- ? = ユーザ定義信号名
- 以下の場合には
  - A2 = T の場合、X1 = 別のモジュールへの出力信号となります
  - A2 = F の場合、X1 = 別のモジュールからの入力信号となります
  - X1 信号名は、A1 の値 + アンダースコア (_) + X1 の名前です 信号名はグローバルになります

有効接続

ピン	データ型	ピン	データ型
A1	TL	X1	ALL
A2	BOOL		

例: Configurable Module Connection

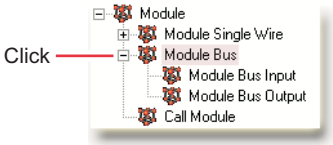


A_ADD signal outputs from Module 1

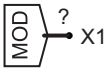
A_ADD signal inputs to Module 2

When placing or moving components, press F8 to reflect them180°

Module Bus（モジュールバス）



Module Bus Input（モジュールバス入力）



用途

- モジュールバスのための入力。

機能

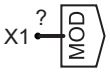
- ? = ユーザ定義バス名
- X1 = バス出力

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

本コンポーネントの使い方は*308*ページの *例 2: Call Module*を参照してください。

Module Bus Output（モジュールバス出力）



用途

- モジュールバスのための出力

機能

- ? = ユーザ定義バス名
- X1 = バス入力

有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

本コンポーネントの使い方は***308***ページの *例 **2: Call Module***を参照してください。

Call Module（モジュール呼び出し）

用途

- コールのパラメータなしに、別のファイルに含まれているモジュールを実行します。

機能

- NAME ?をユーザ定義モジュール名に置き換えます

  入力される名前は、**Module.scs** ファイルの **Module name** に一致しなければなりません

  （**Module name** は **Module.scs** ファイルのプロパティです。**Project Manager** タブの **Module.scs** ファイルをクリックすると、**Inspector** タブに **Module name** が表示されます。）

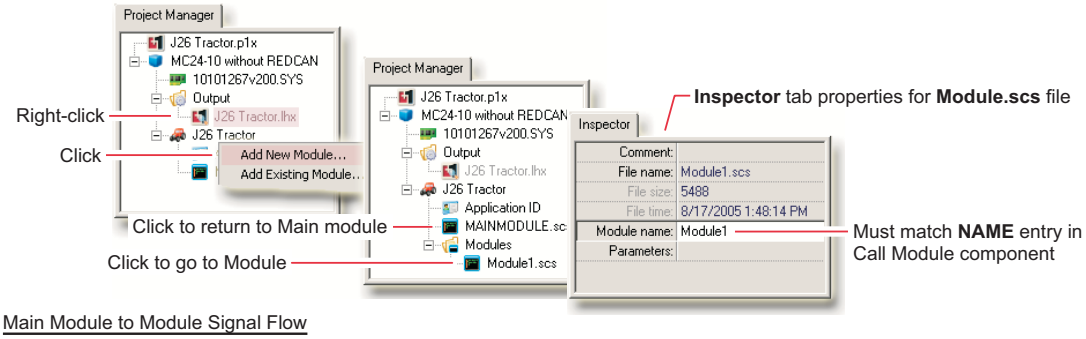
- COMMENT をユーザコメント文字列に置き換えます
- 以下の場合には
  - A1 = T の場合、NAME ?で識別されるモジュール内のコンポーネントが実行されます
  - A1 = F の場合、NAME ?で識別されるモジュール内のコンポーネントは実行されません
- メインループで同じ関数を 2 回以上コールしません
- モジュールがコールされない場合、モジュール出力値は設定されたままとなります
- A1 が T から F に移行すると、いかなる出力値も適用されます。また、A1 が F から T に移行する場合も同様です

本コンポーネントの使い方は以下の*例 1:Call Module*と***308***ページの*例* ***2: Call Module***を参照してください。

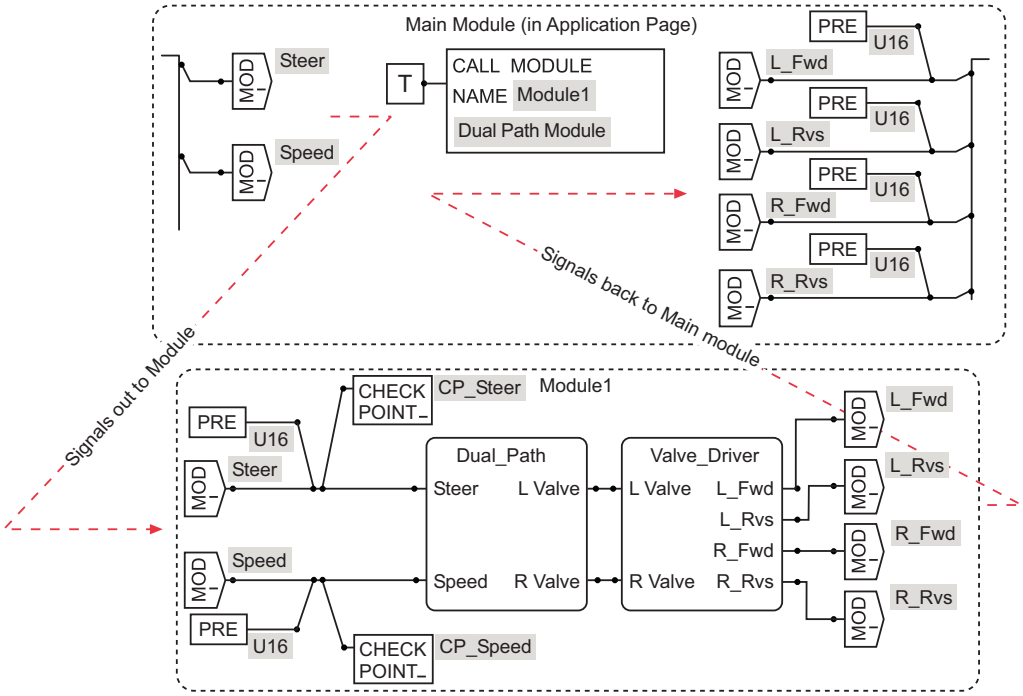
例 1: Call Module

以下の図に、**Module Input Typed**、**Module Output**、**Call Module** コンポーネントを用いてモジュールを作成する方法を示しています。

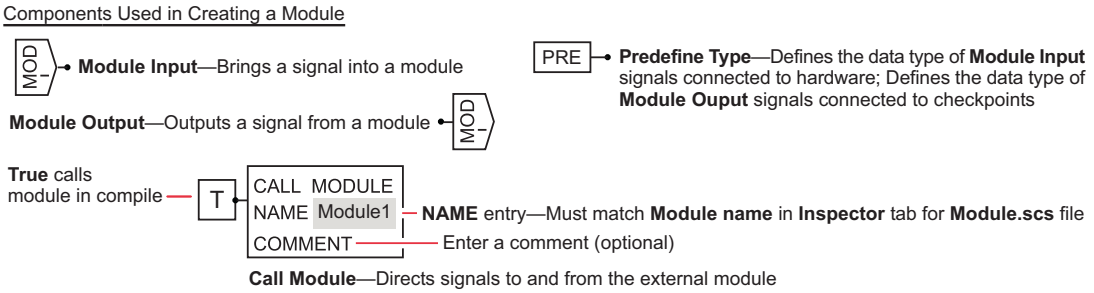
How to Add a Module



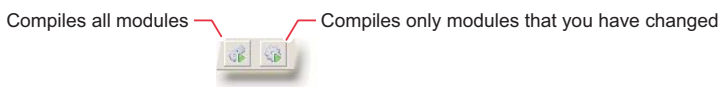
Main Module to Module Signal Flow



Components Used in Creating a Module



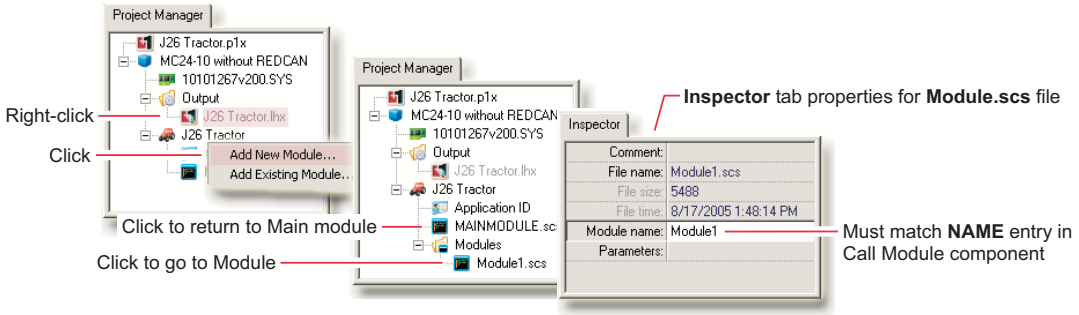
Compile Buttons in the Toolbar



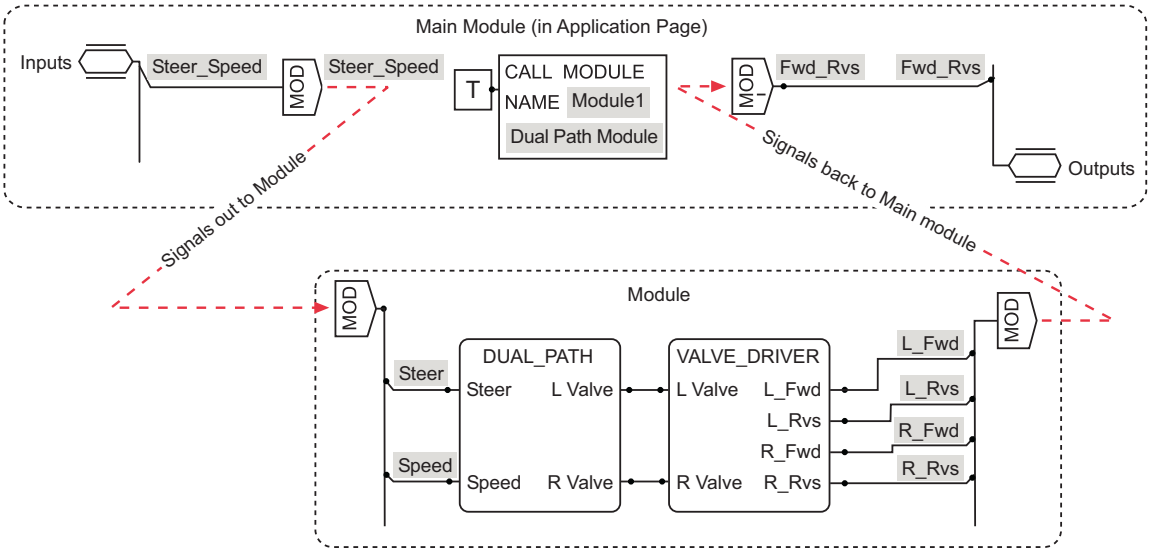
例 2: Call Module

以下の図に、**Module Bus Input**、**Module Output Bus**、**Call Module** コンポーネントを用いてモジュールを作成する方法を示しています。

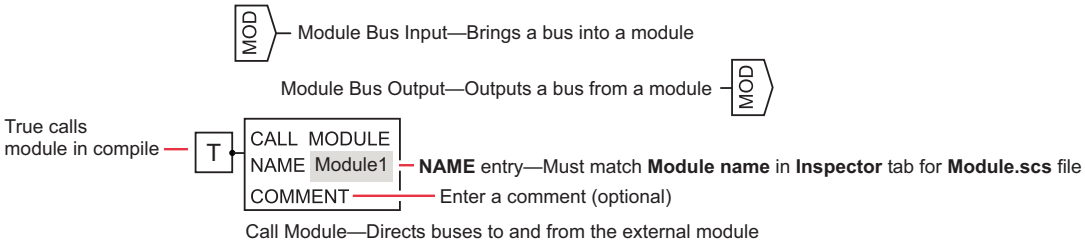
How to Add a Module



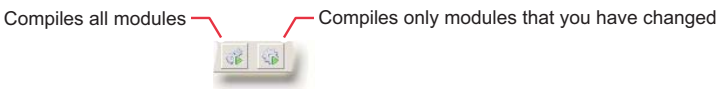
Main Module to Module1 Signal Flow



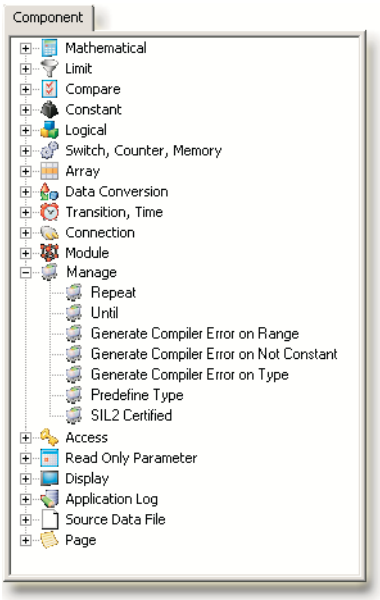
Components Used in Creating a Module



Compile Buttons in Toolbar



Manage（管理）


Component
Mathematical
Limit
Compare
Constant
Logical
Switch, Counter, Memory
Array
Data Conversion
Transition, Time
Connection
Module
Manage
Repeat
Until
Generate Compiler Error on Range
Generate Compiler Error on Not Constant
Generate Compiler Error on Type
Predefine Type
SIL2 Certified
Access
Read Only Parameter
Display
Application Log
Source Data File
Page

Repeat

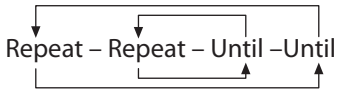
用途

- 単一のアプリケーションループ内でコンポーネントを繰り返し実行します。
- Repeat-Until ループの始点をマークします。**Until** コンポーネントがループの終点をマークします。
- 配列を読み込んだり書き込んだりするときに便利です。

機能

- コンポーネントの左側の点が Repeat-Until ループの始点をマークします。
- **Until**コンポーネントと一対で使用されます（***311***ページの***Until***を参照）。

  Repeat と **Until** コンポーネントは同一ページに置く必要があります。PLUS+1 GUIDE プログラムでは複数ページにまたがる Repeat-Until ループは使えません。
- 各アプリケーションループ内の現在の Repeat-Until ループの最初の繰り返しの間、X1 = T となり、その後 F となります。Repeat-Until ループが最初の繰り返しの直後に終了した場合はこの値は Repeat-Until ループの外でも T のまま残ります。
- Repeat-Until を入れ子にすることも可能です。



- Repeat-Untilループ内で使用できないアイテムがあります。***312***ページの***Do Not Use Inside Repeat-Until Loops***を参照してください。

有効接続

ピン	データ型	ピン	データ型
—	—	X1	BOOL

Until

用途

- 単一のアプリケーションループ内でコンポーネントを繰り返し実行します。
- Repeat-Until ループの終点をマークします。**Repeat** コンポーネントがループの始点をマークします。

機能

- **Repeat**コンポーネントと一対で使用されます（*310*ページの*Repeat*を参照）。

  Repeat と **Until** コンポーネントは同一ページに置く必要があります。PLUS+1 GUIDE プログラムでは複数ページにまたがる Repeat-Until ループは使えません。
- A1 = T—Repeat-Until ループを終了させます
- タイムアウト条件で Repeat-Until ループが中断したとき X1 = T となります。
  - コントローラは **Until** コンポーネントが実行される度にタイムアウト条件をチェックします。
  - アプリケーションループ時間がコントローラの **OS.ExecTimeOut** を越えた場合にタイムアウト条件が発生します。
  - タイムアウト条件は Repeat-Until ループを中断します。後の Repeat-Until ループは一度実行されたあと中断します。

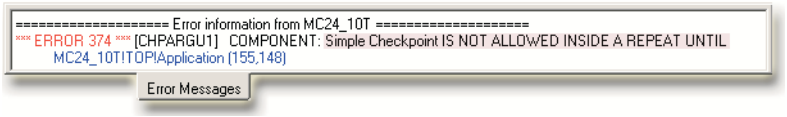
    （タイムアウト条件は、コントローラが Repeat-Until ループの最後の繰り返しで **OS.ExecTimeOut** に達した場合にも発生します。Repeat-Until ループが完全に実行されても、**Until** コンポーネントの X1 出力が T になればタイムアウト条件とみなされます。）
- Repeat-Until を入れ子にすることも可能です。

- Repeat-Untilループ内で使用できないアイテムがあります。*312*ページの*Do Not Use Inside Repeat-Until Loops*を参照してください。

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	INT

Do Not Use Inside Repeat-Until Loops



- オブジェクトページを Repeat-Until ループ内で使用するとコンパイルエラーの原因となります。
- この表のコンポーネントを Repeat-Until ループ内で使用するとコンパイルエラーの原因となります。

これらのコンポーネントを Repeat-Until ループ内で使用しないでください

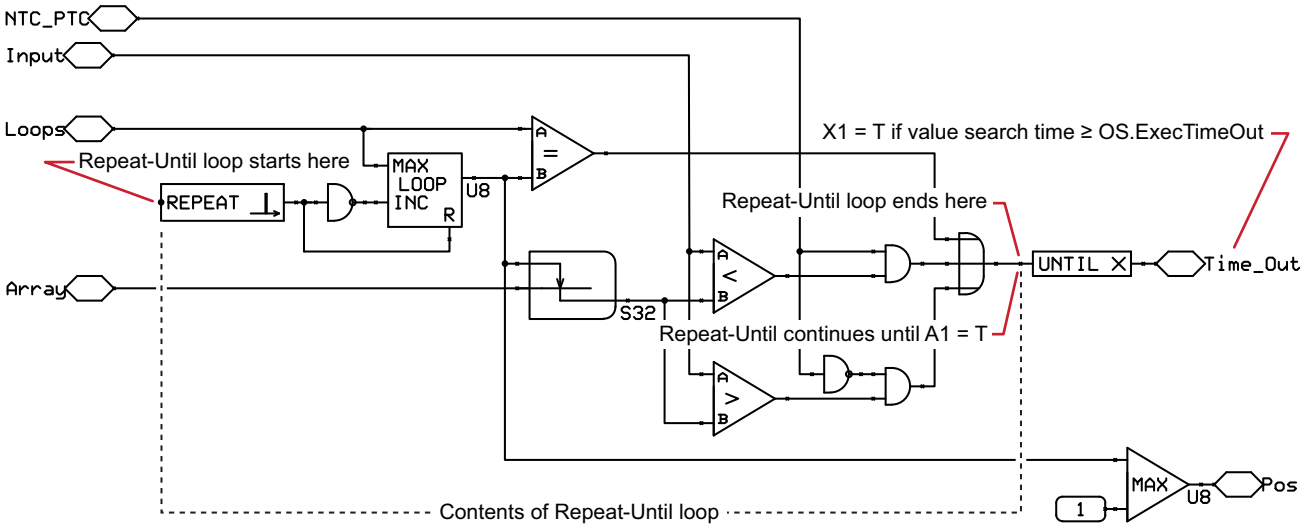
Category	Component Name
Transition, Time	On Delay
	Off Delay
	Oscillator
	Pulse
	Time Base
	Measure Period
	Get Time µs
Connection	Simple Checkpoint
	Advanced Checkpoint
	Advanced Checkpoint with Namespace
	Set Value
	Set Pulse
	Initialize Hardware Output
	Hardware Input Typed
	Hardware Input
	Hardware Output
	Hardware Input/Output Typed
	Hardware Input/Output
	Read Output from Hardware Typed
	Read Output from Hardware
	Non-Volatile Memory Input Typed
	Non-Volatile Memory Input/Output
	Using Old CAN and Memory Components
	Transmit CAN
	Receive CAN with Filter
	Receive CAN with ID Mask
	Receive CAN Basic
	Non-Volatile Memory Dynamic with Default

これらのコンポーネントを Repeat-Until ループ内で使用しないでください

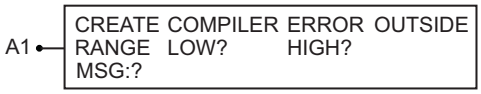
Category	Component Name
	Non-Volatile Memory Dynamic
Module	Module Input
	Module Input Typed
	Module Output
	Configurable Module Connection
	Module Bus Input
	Module Bus Output
	Call Module
Access	Access App Log Enable
	Disable Raw Applog Data Readout
	Accessrights Diag Access
	Accessrights App Log Errors
	Accessrights App Log Others
Display	Define Window
	Line
	Graphic
	Text Label
	Graphic Label
	String
	String with 4 Variables
	Basic String with 4 Variables
	Basic String with 8 Variables
	Open Text Set
	Close Text Set
	Language Definition Input
	Select Language
	Define Areas Page
	Define Screen Page
Application Log	Language Definition Input
	Select Language
	Define Application Log Areas Page
	Define Application Log Page

例: Repeat-Until

Search for the closest value in an array



Generate Compiler Error on Range（範囲に関するコンパイラエラーの生成）



用途

- コンパイラが範囲外の定数値を検出すると、コンパイルプロセスを打ち切ります。
- **Error Messages**（エラーメッセージ）タブにユーザ定義エラーメッセージを表示します。
- コンパイル可能だがダウンロード時に誤動作を引き起こす値をユーザが入力するのを防ぎます。

  （例えば、ジョイスティック較正値が高すぎる場合でもコンパイルは成功しますが、ダウンロード時に誤動作を引き起こすことがあります。）

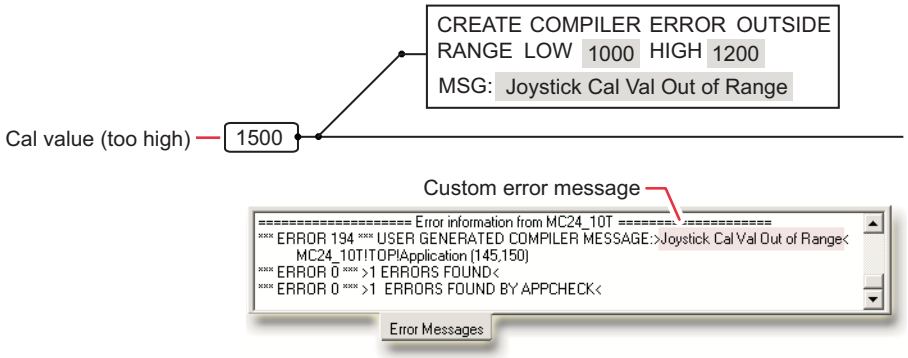
機能

- A1 = 定数値
- Low? = Query/Change でクリックして範囲の低い方の値を入力します
- High? = Query/Change でクリックして範囲の高い方の値を入力します
- MSG? = Query/Change でクリックして A1 が範囲外である場合に表示するエラーメッセージを入力します

有効接続

ピン	データ型	ピン	データ型
A1	INT	—	—

例: Generate Compiler Error on Range



Generate Compiler Error on Not Constant（定数でないことに関するコンパイラエラーの生成）

用途

- コンパイラが定数値ではなく変数値を検出すると、コンパイルプロセスを打ち切ります。
- **Error Messages**（エラーメッセージ）タブにユーザ定義エラーメッセージを表示します。
- コンパイルするがダウンロード時に誤動作を引き起こす値をユーザが入力するのを防ぎます。

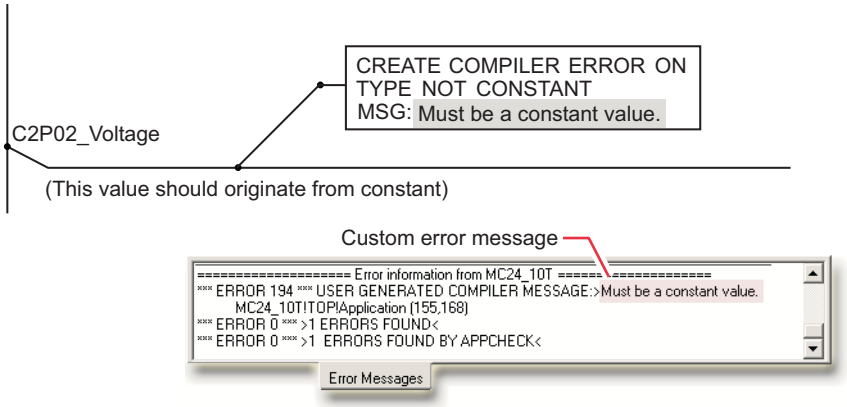
機能

- A1 = 定数データ型（他のデータ型ではコンパイルが打ち切られます）
- MSG? = Query/Change でクリックして A1 が定数値ではなく変数値である場合に表示するエラーメッセージを入力します

有効接続

ピン	データ型	ピン	データ型
A1	ALL	—	—

例: Generate Compiler Error on Not Constant



Generate Compiler Error on Type（データ型に関するコンパイラエラーの生成）

用途

- コンパイラが誤ったデータ型を検出すると、コンパイルプロセスを打ち切ります。
- **Error Messages** タブにユーザ定義エラーメッセージを表示します。
- コンパイルするがダウンロード時に誤動作を引き起こす値をユーザが入力するのを防ぎます。

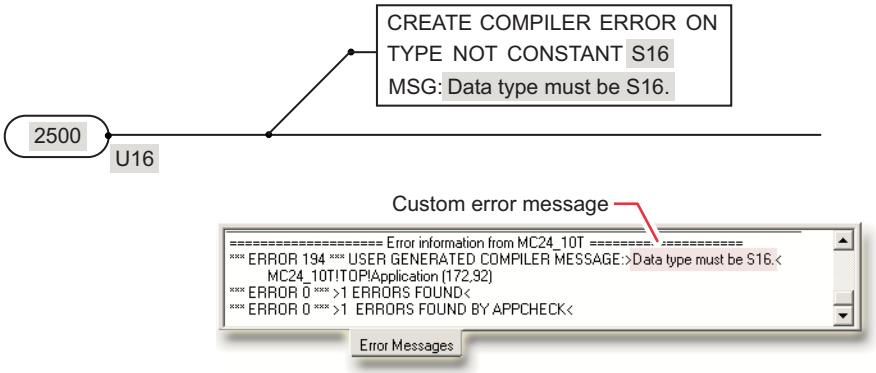
機能

- A1 = データ型
- TYPE DIFFERS FROM? = Query/Change でクリックして許容データ型を入力します（他のデータ型ではコンパイルが打ち切られます）
- MSG? = Query/Change でクリックして A1 が誤ったデータ型である場合に表示するエラーメッセージを入力します

有効接続

ピン	データ型	ピン	データ型
A1	ALL	—	—

例: Generate Compiler Error on Type



Predefine Type（事前定義データ型）

PRE → X1
?

用途

- データ型を持たない信号のデータ型を定義します。
  - ハードウェア出力に送られる **Module Input** 信号のデータ型を定義する必要があります。
  - **Checkpoint** コンポーネントに送られる **Module Output** 信号のデータ型を定義する必要があります。
- 正しい信号を入力または出力に強制的に接続します。
  - 事前定義信号に送られる信号は、データ型を持たないか、適合するデータ型を持っていなければなりません。そうでない場合は、コンパイラエラーが発生します。

機能

- X1 = ?データ型を出力します

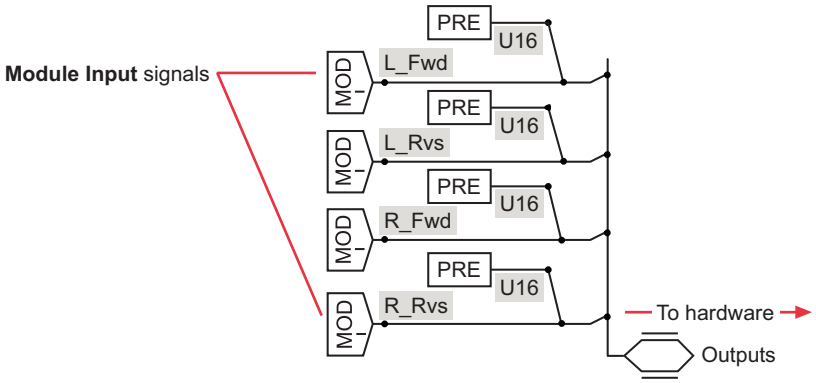
有効接続

ピン	データ型	ピン	データ型
—	—	X1	ALL

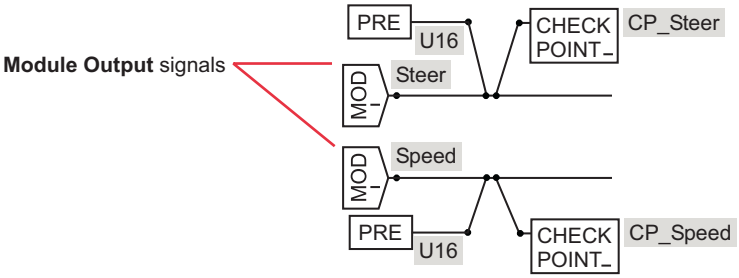
本コンポーネントの使い方は ***307*** ページの *例 **2: Call Module*** を参照してください。

例: Predefine Type

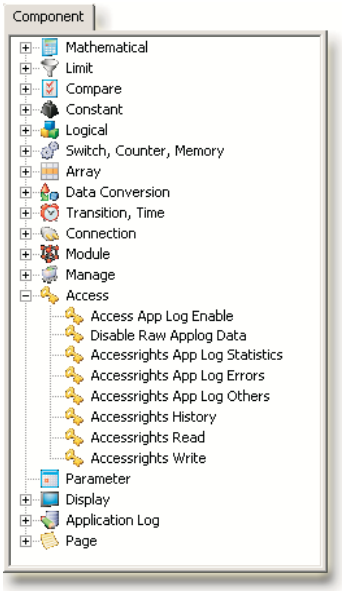
Define the data type of **Module Input** signals routed to hardware



Define the data type of **Module Output** signals routed to **Checkpoints**



Access（アクセス）


Component
Mathematical
Limit
Compare
Constant
Logical
Switch, Counter, Memory
Array
Data Conversion
Transition, Time
Connection
Module
Manage
Access
Access App Log Enable
Disable Raw Applog Data
Accessrights App Log Statistics
Accessrights App Log Errors
Accessrights App Log Others
Accessrights History
Accessrights Read
Accessrights Write
Parameter
Display
Application Log
Page

Access App Log Enable

A1 APP–LOG ENABLE

用途

- パスワードロジックと共に使用し、PLUS+1 サービスツールプログラムのアプリケーションデータログへのアクセス権限を制限します。

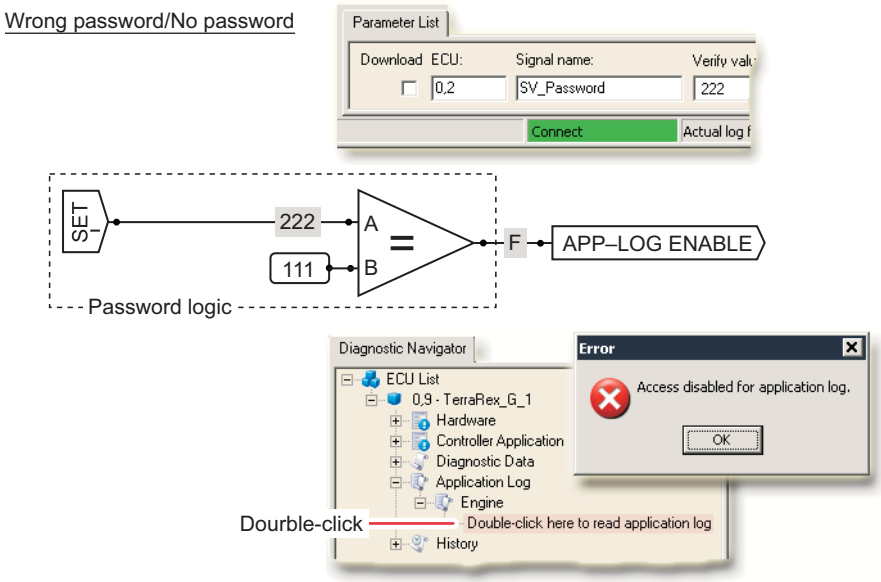
機能

- 以下の場合には
  - A1 = T の場合、サービスツールプログラムはアプリケーションデータログにアクセスできます。
  - A1 = F の場合、サービスツールプログラムはアプリケーションデータログにアクセスできません。

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	X1	—

例: Access App Log Enable



Disable Raw Applog Data Readout

A1 ← DISABLE APPLOG RAW DATA READOUT

用途

- パスワードロジックと共に使用し、PLUS+1 サービスツールプログラムの P1A ファイル（アプリケーションデータログの生データを含むファイル）へのアクセス権限を制限します。

機能

- 以下の場合には、
  - A1 = T の場合、サービスツールプログラムは P1A ファイルにアクセスできません。

    サービスツールプログラムは他の **Access** コンポーネントで許可されたデータ、例えば **App-Log Error Access** コンポーネントを CSV ファイルに書き込むことのみ可能です。
  - A1 = F の場合、サービスツールプログラムは P1A ファイルにアクセスできます。

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	—	—

Accessrights App Log Statistics（アプリケーションログ 統計データへのアクセス権）

A1 APP–LOG STAT ACCESS

用途

- パスワードロジックと共に使用し、アプリケーションデータログに含まれる統計データへのアクセス権限を制限します。
- アプリケーションデータログに書き込まれるデータは以下のユーザ定義を持ちます
  - **S**, **E**, または **O** の文字の **Tag** はデータを統計（Statistical）、エラー（Error）、あるいはその他（Other）として分類します。
  - **Access** レベルを **0～9** の範囲で設定します。

    アプリケーションデータログに関する詳細な情報は ***405*** ページの*アプリケーションデータログ機能*を参照してください。

機能

- A1 の範囲は 0～9 です
- サービスツールプログラムは
  - アプリケーションデータログ内のアクセスレベル ≥ **A1** の統計データにアクセスすることができます
  - アプリケーションデータログ内のアクセスレベル < **A1** の統計データにアクセスすることができません
- アプリケーション内のどこにでも配置できます

有効接続

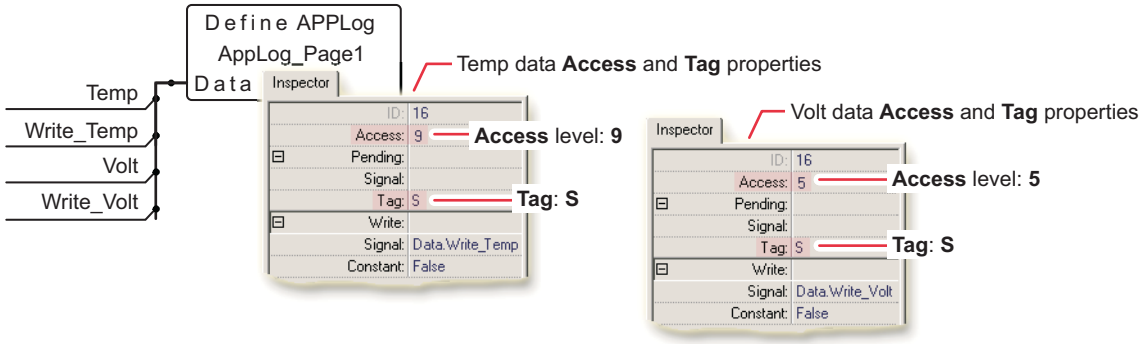
ピン	データ型	ピン	データ型
A1	INT	—	—

Diagnostic data access levels

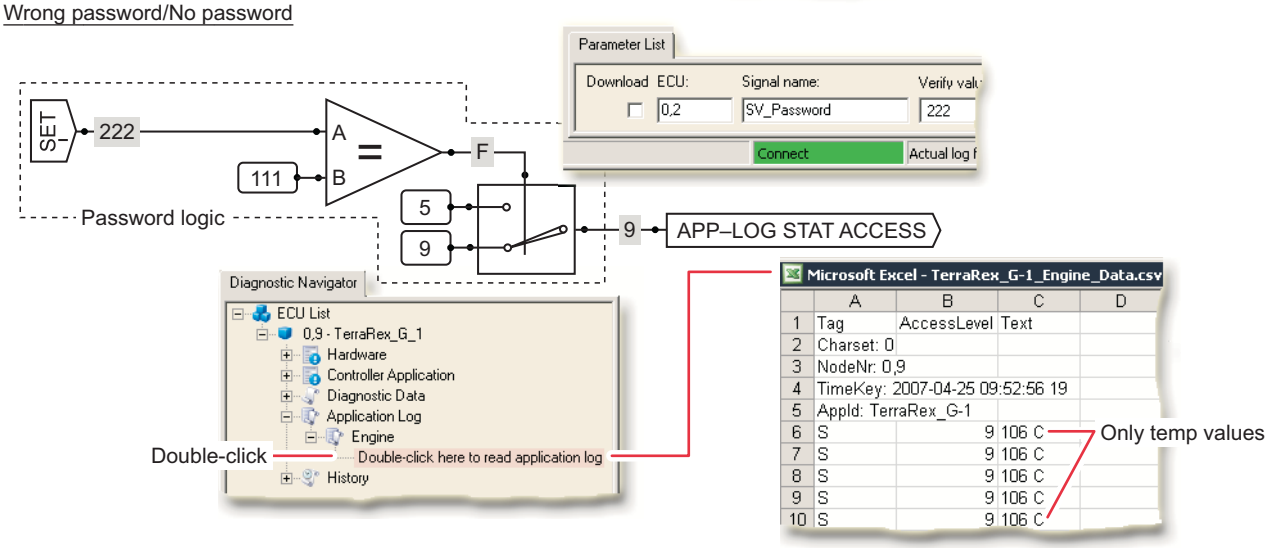
A1 Access Value	Can Access Statistical Data with These Access Levels										
0 APP-LOG STAT ACCESS	0	1	2	3	4	5	6	7	8	9	Most
3 APP-LOG STAT ACCESS				3	4	5	6	7	8	9	
6 APP-LOG STAT ACCESS							6	7	8	9	
9 APP-LOG STAT ACCESS										9	Least

例: Accessrights App Log Statistics

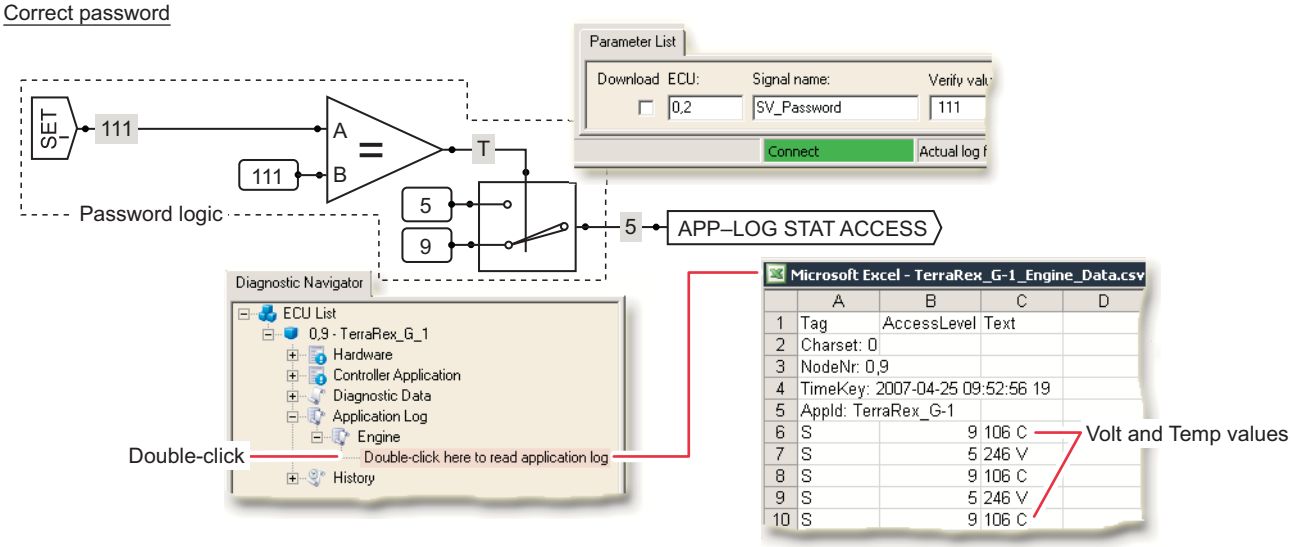
Inspector tab inside **Define AppLog** component



Wrong password/No password



Correct password



Accessrights App Log Errors（アプリケーションログ エラーデータへのアクセス権）

A1 APP–LOG ERROR ACCESS

用途

- アプリケーションデータログ内のエラーデータへの PLUS+1 サービスツールプログラムのアクセス権をパスワードロジックを使って設定します。
- アプリケーションデータログに書き込まれるデータは以下のユーザ定義を持ちます
  - **S**, **E**, または **O** の文字の **Tag** はデータを統計（Statistical）、エラー（Error）、あるいはその他（Other）として分類します。
  - **Access** レベルを **0～9** の範囲で設定します。

    アプリケーションデータログに関する詳細な情報は ***405*** ページの*アプリケーションデータログ機能*を参照してください。

機能

- A1 の範囲は 0～9 です
- サービスツールプログラムは
  - アプリケーションデータログ内のアクセスレベル ≥ A1 のエラーデータにアクセスすることができます
  - アプリケーションデータログ内のアクセスレベル < A1 のエラーデータにアクセスすることができません
- アプリケーション内のどこにでも配置できます

有効接続

ピン	データ型	ピン	データ型
A1	INT	—	—

Error data access levels

A1 Access Level	Can Access Error Data with These Access Levels										
0 APP-LOG ERROR ACCESS	0	1	2	3	4	5	6	7	8	9	Most
3 APP-LOG ERROR ACCESS				3	4	5	6	7	8	9	
6 APP-LOG ERROR ACCESS							6	7	8	9	
9 APP-LOG ERROR ACCESS										9	Least

APP-LOG STATS ACCESSコンポーネントは**APP-LOG ERROR ACCESS**コンポーネントと同様、アプリケーションデータログに含まれるデータの型へのアクセスを制限します。***324*** ページの*例: **Accessrights App Log Statistics***を参照してください。

Accessrights App Log Others（アプリケーションログ その他のデータへのアクセス権）

A1 APP–LOG OTHERS ACCESS

用途

- アプリケーションデータログ内の「その他」（miscellanous）データへの PLUS+1 サービスツールプログラムのアクセス権をパスワードロジックを使って設定します。
- アプリケーションデータログに書き込まれるデータは以下のユーザ定義を持ちます
  - **S**, **E**, または **O** の文字の **Tag** はデータを統計（Statistical）、エラー（Error）、あるいはその他（Other）として分類します。
  - **Access** レベルを **0～9** の範囲で設定します。

    アプリケーションデータログに関する詳細な情報は ***405*** ページの *アプリケーションデータログ機能*を参照してください。

機能

- A1 の範囲は 0～9 です
- サービスツールプログラムは
  - アプリケーションデータログ内のアクセスレベル ≥ **A1** の「その他」データにアクセスすることができます
  - アプリケーションデータログ内のアクセスレベル < **A1** の「その他」データにアクセスすることができません
- アプリケーション内のどこにでも配置できます

有効接続

ピン	データ型	ピン	データ型
A1	INT	—	—

Other data access levels

A1 Access Value	Can Access Other Data with These Access Levels										
0 APP-LOG OTHERS ACCESS	0	1	2	3	4	5	6	7	8	9	Most
3 APP-LOG OTHERS ACCESS				3	4	5	6	7	8	9	
6 APP-LOG OTHERS ACCESS							6	7	8	9	
9 APP-LOG OTHERS ACCESS										9	Least

APP-LOG STATS ACCESS コンポーネントは**APP-LOG OTHERS ACCESS** コンポーネントと同様、アプリケーションデータログに含まれるデータの型へのアクセスを制限します。***324***ページの *例: **Accessrights App Log*** Statisticsを参照してください。

Accessrights History（履歴データへのアクセス権）

A1 HISTORY ACCESS

用途

- パスワードロジックと共に使用し、アプリケーションデータログ内のアクセス履歴データへのPLUS+1 サービスツールプログラムのアクセス権を設定します。

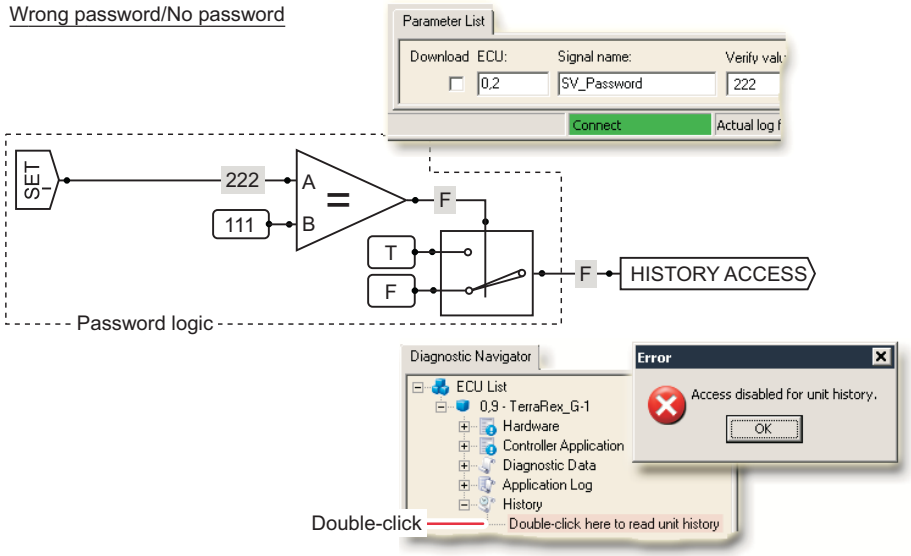
機能

- サービスツールプログラムは
  - A1= T のアクセス履歴データを閲覧することができます
  - A1= F のアクセス履歴データを閲覧することができません
- アプリケーション内のどこにでも配置できます
- サービスツールプログラムで設定したアクセス権より優先します。

有効接続

ピン	データ型	ピン	データ型
A1	BOOL	—	—

例: Accessrights History



Accessrights Read（読取アクセス権）

A1 READ ACCESS

用途

- パスワードロジックと共に使用し、コンポーネント値への PLUS+1 サービスツールプログラムからの読取アクセス権を設定します。

  PLUS+1 コンポーネントの中には、**Checkpoint** コンポーネントのように、読取アクセスレベル機能のあるものがあります。これらの読取アクセスレベルの範囲は 0～9 です。これらの読取アクセスレベルの範囲は 0～9 です。

  Read Access コンポーネントの **A1** 値は、個々のコンポーネント読取アクセスレベルと連動して、サービスツールがパスワードなしで読み取ることのできる値を制限します。

機能

- A1 の範囲は 0～9 です
- サービスツールプログラムは
  - アクセスレベル≥A1 のコンポーネントの値を読み取ることができます
  - アクセスレベル<A1 のコンポーネントの値を読み取ることができません
- アプリケーション内のすべてのコンポーネントの読取アクセス権を設定します
- アプリケーション内のどこにでも配置できます
- サービスツールプログラムで設定した読み取りアクセスレベルより優先します。

有効接続

ピン	データ型	ピン	データ型
A1	INT	—	—

Read access levels

A1 Value*	Can Read Values from Components with These Read Access Levels										
0 READ ACCESS	0	1	2	3	4	5	6	7	8	9	Most
3 READ ACCESS				3	4	5	6	7	8	9	
6 READ ACCESS							6	7	8	9	
9 READ ACCESS										9	Least

*Typically set with password logic.

Accessrights Write（書込アクセス権）

A1 WRITE ACCESS

用途

- パスワードロジックと共に使用し、PLUS+1 サービスツールプログラムにコンポーネントへの書込アクセス権を設定します

  PLUS+1 コンポーネントの中には、**Memory** コンポーネントのように、書込アクセスレベル機能のあるものがあります。これらの書込アクセスレベルの範囲は 0～9 です。

  Write Access コンポーネントの **A1** 値は、個々のコンポーネント書込アクセスレベルと連動して、サービスツールがパスワードなしで変更する(書き込む)ことのできる値を制限します。

機能

- A1 の範囲は 0～9 です
- サービスツールプログラムを使用し、ユーザは
  - 書き込みアクセスレベル ≥ A1 のコンポーネントに値を書き込むことができます
  - 書き込みアクセスレベル <A1 のコンポーネントに値を書き込むことができません
- アプリケーション内のすべてのコンポーネントの書込アクセス権を設定します
- アプリケーション内のどこにでも配置できます
- サービスツールプログラムで設定した書き込みアクセスレベルより優先します。

有効接続

ピン	データ型	ピン	データ型
A1	INT	—	—

Write access levels

A1 Value*	Can Write Values to Components with These Write Access Levels										
0 WRITE ACCESS	0	1	2	3	4	5	6	7	8	9	Most
3 WRITE ACCESS				3	4	5	6	7	8	9	
6 WRITE ACCESS							6	7	8	9	
9 WRITE ACCESS										9	Least

*Typically set with password logic.

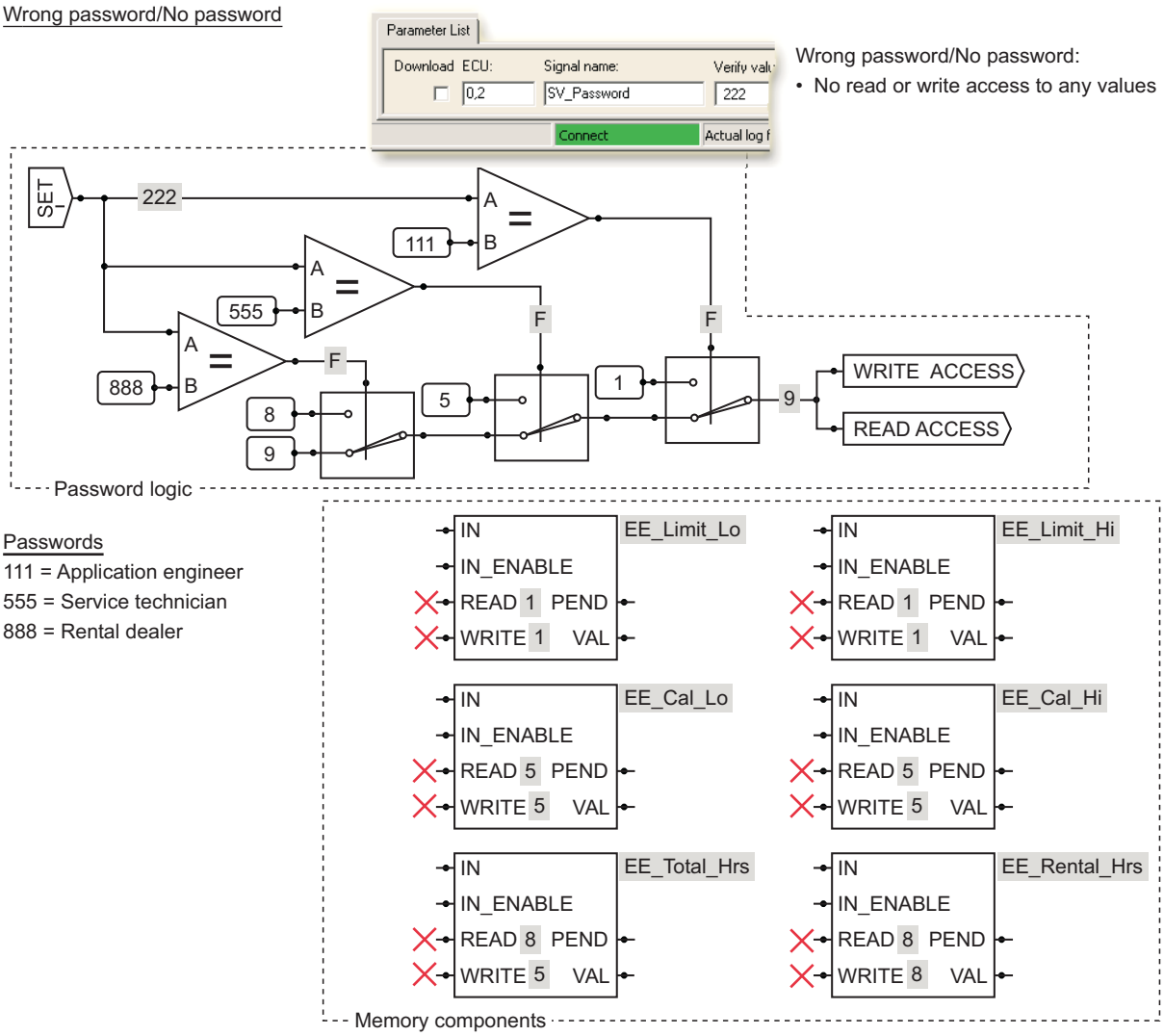
例: Read Access and Write Access

このあとの例では **Read Access** コンポーネントと **Write Access** コンポーネントと共に使用されるパスワードロジックが、どのように PLUS+1 サービスツールプログラムのコンポーネントの値の読み取りおよび書き込み権限を制限するかを示しています。

以下の例は、サービスツールプログラムの使用者が正しくないパスワードを入力したりパスワードを入力しない場合です。

パスワードロジックは **9** という値を **Write Access** コンポーネントと **Read Access** コンポーネントに適用します。サービスツールプログラムは書き込み/読み取りアクセスレベルが **9** の **Memory** コンポーネントのみ読み取りや書き込みができます。（ここではアクセスレベル **9** のコンポーネントはありません。）

パスワードなし、または正しくないパスワードを入力したユーザはどの値も確認したり変更したりできません。

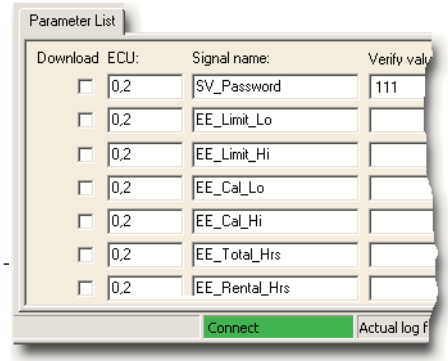


以下の例はアプリケーションエンジニアが PLUS+1 サービスツールを使ってパスワード **111** を入力したときの動作を示しています。

パスワードロジックにより **Write Access** コンポーネントと **Read Access** コンポーネントに値 **1** を適用します。この値によりサービスツールプログラムは読み取り/書き込みアクセスレベルが **1** 以上の全ての **Memory** コンポーネントへの読み取りおよび書き込みが行えます。

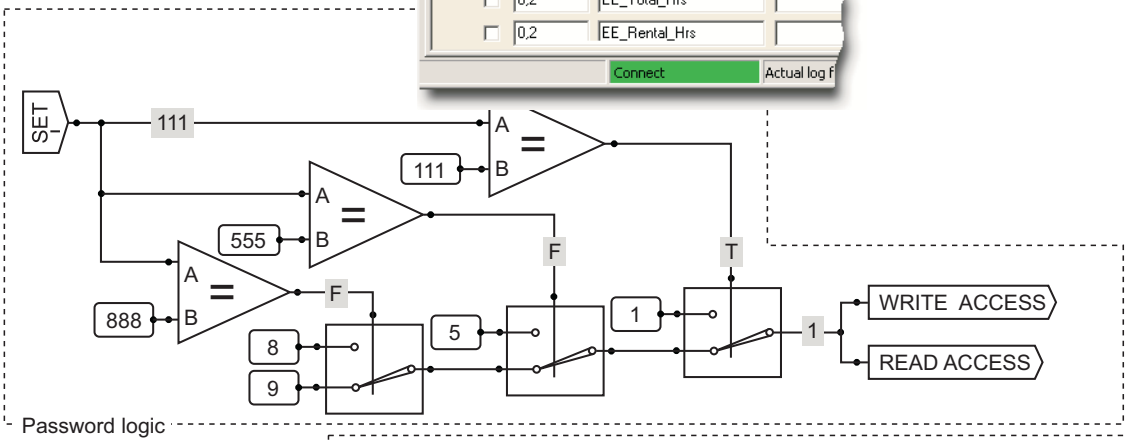
アプリケーションエンジニアは全ての値を確認したり変更したりできます。

Application engineer password



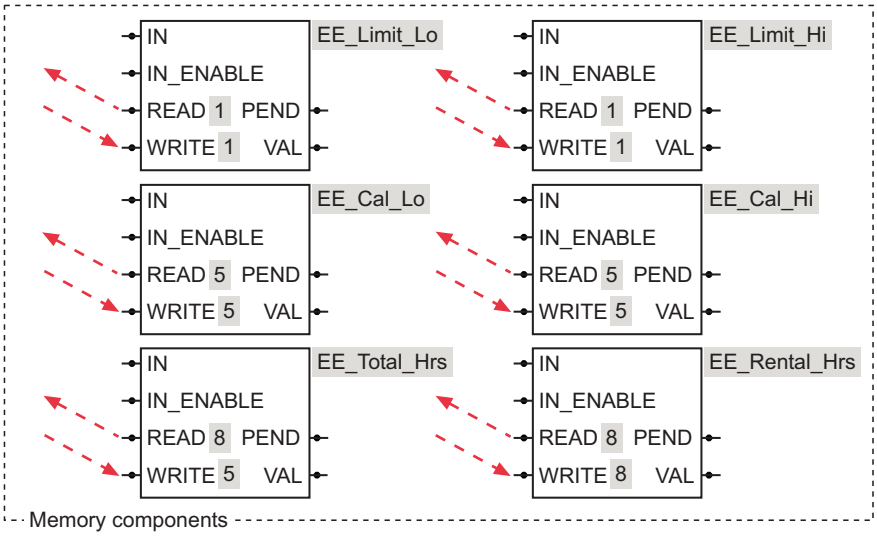
Application engineer password:

- Read and write access to **EE_Limit_Lo** and **EE_Limit_Hi** values
- Read and write access to **EE_Cal_Lo** and **EE_Cal_Hi** values
- Read and write access to **EE_Total_Hrs** and **EE_Rental_Hrs** values



Password logic

Passwords
111 = Application engineer
555 = Service technician
888 = Rental dealer



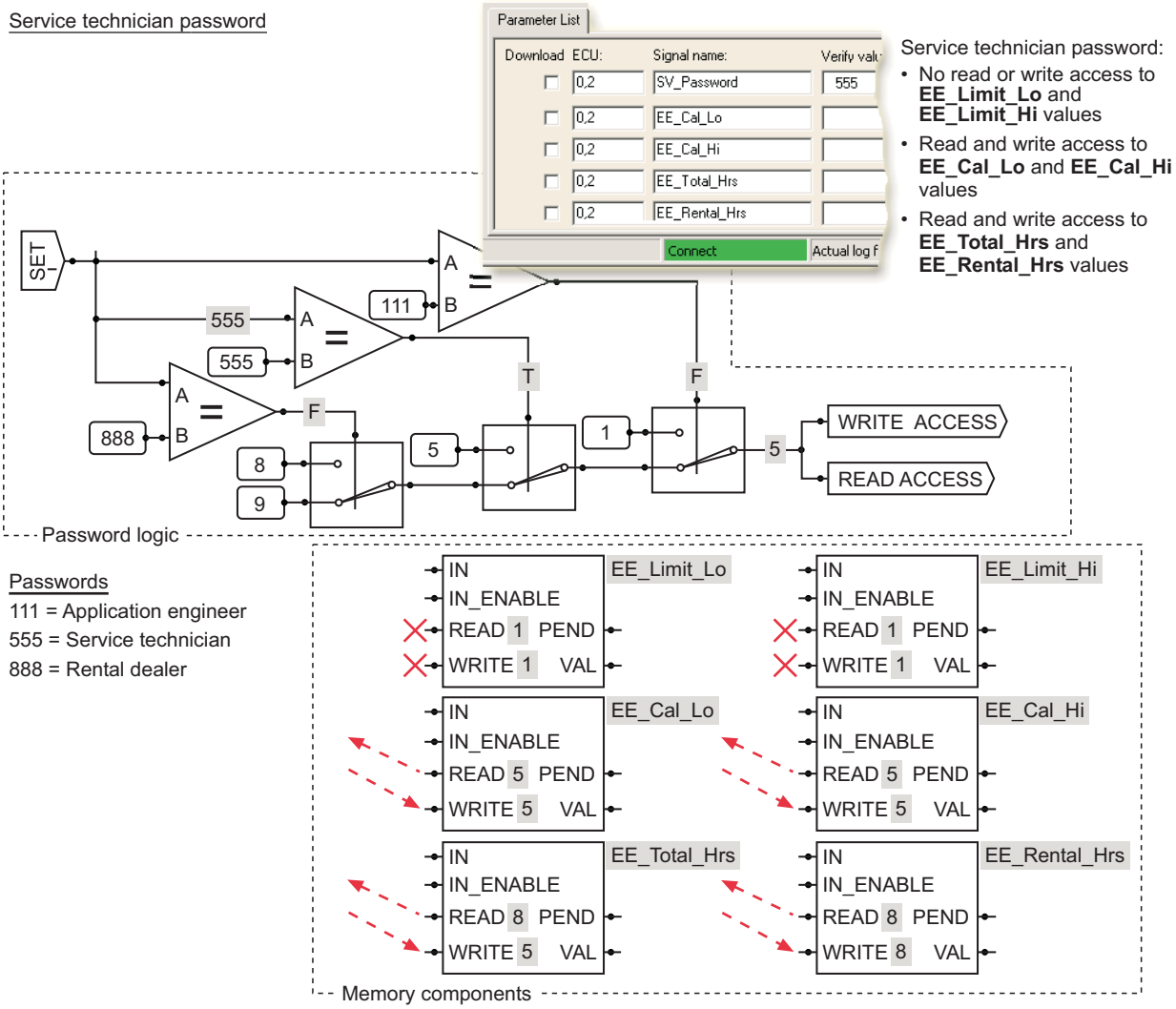
Memory components

以下の例はサービスエンジニアが PLUS+1 サービスツールを使ってパスワード **555** を入力したときの動作を示しています。

パスワードロジックにより **Write Access** コンポーネントと **Read Access** コンポーネントに値 **5** を適用します。この値によりサービスツールプログラムは読み取り/書き込みアクセスレベルが **5** 以上の全ての **Memory** コンポーネントへの読み取りおよび書き込みが行えます。

サービスエンジニアは、

- リミット値を確認したり変更したりできません
- キャリブレーション値を確認したり変更したりできます
- 総稼働時間を確認したり変更（リセット）したりできます
- レンタル時間を確認したり変更（リセット）したりできます

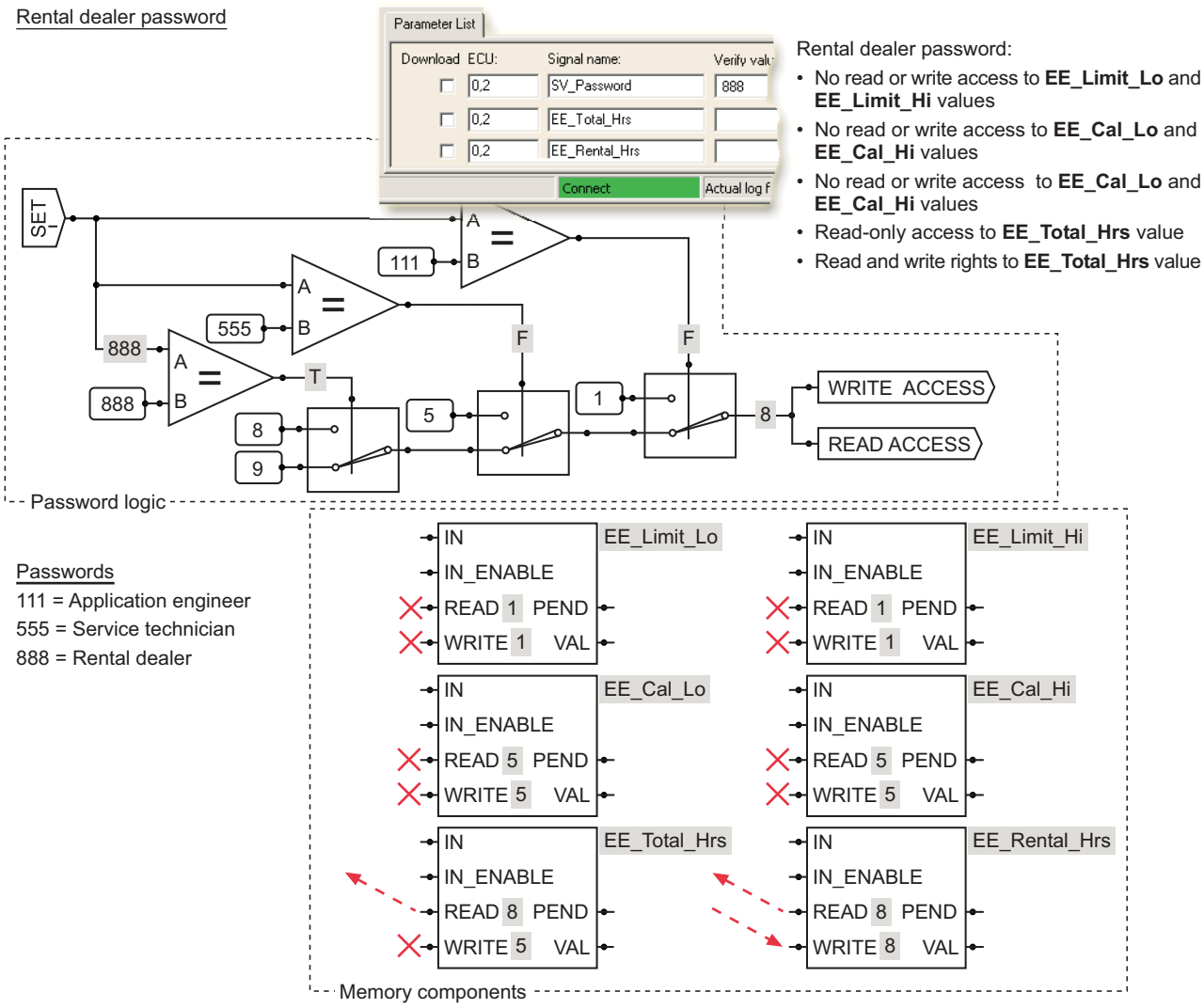


以下の例はレンタル会社のユーザが PLUS+1 サービスツールを使ってパスワード **888** を入力したときの動作を示しています。

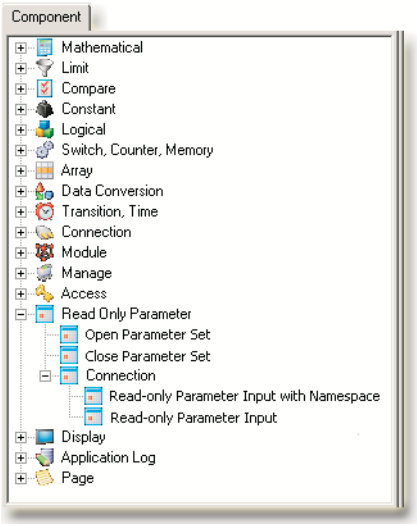
パスワードロジックにより **Write Access** コンポーネントと **Read Access** コンポーネントに値 **8** を適用します。この値によりサービスツールプログラムは読み取り/書き込みアクセスレベルが **8** 以上の全ての **Memory** コンポーネントへの読み取りおよび書き込みが行えます。

レンタル会社は、

- リミット値を確認したり変更したりできません
- キャリブレーション値を確認したり変更したりできません
- 総稼働時間を確認することはできますが変更（リセット）することはできません
- レンタル時間を確認したり変更（リセット）したりできます



Read-only Parameter（読取専用パラメータ）



Open Parameter Set

用途

- 読み取り専用パラメータ入力の開始点をマークします

機能

- 読み取り専用パラメータ lhx ファイル内のパラメータがアプリケーション内のパラメータに一致したときは X1 = T
  - TYPE 名、パラメータ名、パラメータの順序、データ型は一致していなければなりません
  - このコンポーネントの T 出力が、読み取り専用パラメータ lhx ファイル内の値をコントローラに書き込むようにプログラムします
- 読み取り専用パラメータ lhx ファイル内のパラメータがアプリケーション内のパラメータに一致しないときは X1 = F
  - このコンポーネントからの F 出力を扱うようにアプリケーションをプログラムします
- TYPE はパラメータセットとして分かりやすい名前に変更します

*339*ページの *例: **Read-only Parameter Input***を参照してください。また *453*ページの ***Create a Read-only Parameters File t***も参照してください。

有効接続

ピン	データ型	ピン	データ型
—	INT	X1	BOOL

Close Parameter Set

PARAMETER CLOSE
COMMENT

用途

- 読み取り専用パラメータの終点をマークします。

機能

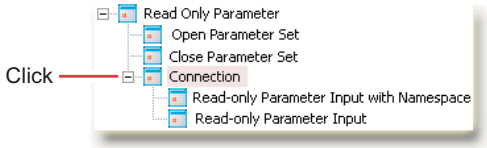
- COMMENT = 分かりやすい名前に変更します

*339*ページの *例*: ***Read-only Parameter Input***を参照してください。また *453*ページの ***Create a Read-only Parameters File*** *t*も参照してください。

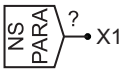
有効接続

ピン	データ型	ピン	データ型
—	—	—	—

Connection



Read-only Parameter Input with Namespace



用途

- 読み取り専用パラメータをアプリケーションに入力します
- パラメータセットをコピーするときはブロックの Name Space を変更するだけでよく、個別の ? パラメータ名を変更する必要はありません。

機能

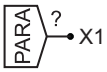
- ブロックの Name Space をプレフィックスとして ? パラメータ名に付けます
- ? = パラメータ名
- X1 = 読み取り専用パラメータ lhx ファイルから入力されたパラメータ値

339ページの *例*: ***Read-only Parameter Input***を参照してください。また *453*.ページの ***Create a Read-only Parameters File t***も参照してください。

有効接続

ピン	データ型	ピン	データ型
—	—	X1	INT, BOOL

Read-only Parameter Input



用途

- アプリケーションに読み取り専用パラメータを入力します

機能

- ? = パラメータ名
- X1 = 読み取り専用 lhx ファイルから入力されたパラメータ値

*339*ページの *例: **Read-only Parameter Input***を参照してください。また *453*.ページの ***Create a Read-only Parameters File** t*も参照してください。

有効接続

ピン	データ型	ピン	データ型
—	—	X1	INT, BOOL

例: Read-only Parameter Input

この例では以下の様子を示しています。

- 読み取り専用パラメータ lhx ファイルのソースファイルを Microsoft Excel で開いたところ
- **PARAMETER OPEN** コンポーネントが、有効な読み取り専用パラメータ lhx ファイルに対して **T** を出力し、無効な読み取り専用パラメータ lhx ファイルに対して **F** を出力しているところ
- **Switch 2** コンポーネントへの **T** 入力によりこれらの出力がそれぞれ読み取り専用パラメータ値の **4500**、**7500**、**T** となっているところ
- **Switch 2** コンポーネントへの **F** 入力によりそれらの出力が内部値の **0**、**0**、**F** となっているところ。

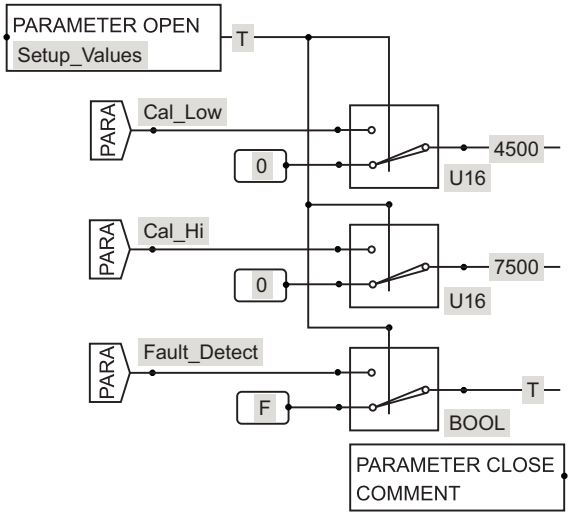
Valid Read-only Parameter lhx File—**PARAMETER OPEN** X1 = T

Comma-separated surce file for the read-only parameters lhx file

Microsoft Excel - DataTemplate_SETUP_VALUES_1.csv

	A	B	C
1	Format;;001A;		
2	Type;;SETUP_VALUES;		
3	TypeFileName;;ROPARAMETER.EXR;		
4	OutputFileName;;DataTemplate_SETUP_V		
5	VERSION;;0;		
6	ADDRESSMODE;;LSBFIRST		
7	DEFAULTTYPEDATA;;1		
8	MIN_DATASIZE;;16		
9	----------;;;		
10	NAME;SIGNALTYPE;VALUE;COMMENT		
11	Cal_Low;U16;4500;		
12	Cal_Hi;U16;7500;		
13	Fault_Detect;BOOL;1;		

Names—OK
Data types—OK
Order—OK

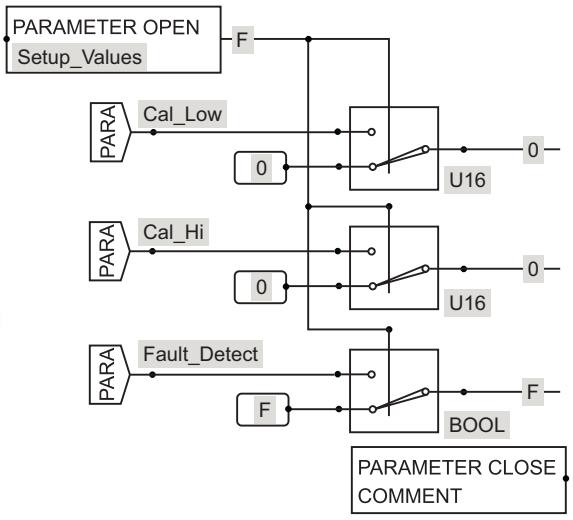


Invalid Read-only Parameter lhx File—PARAMETER OPEN X1 = F

Microsoft Excel - DataTemplate_SETUP_VALUES_1.csv

	A	B	C
1	Format;;001A;		
2	Type;;SETUP_VALUES;		
3	TypeFileName;;ROPARAMETER.EXR;		
4	OutputFileName;;DataTemplate_SETUP_V		
5	VERSION;;0;		
6	ADDRESSMODE;;LSBFIRST		
7	DEFAULTTYPED Cal_Low;U16;4500;		
8	MIN_DATASIZE;;16		
9	----------;;;		
10	NAME;SIGNALTYPE;VALUE;COMMENT		
11	Fault_Detect;BOOL;1;		
12	Cal_Hi;U16;7500;		
13	Cal_Low;U16;4500;		

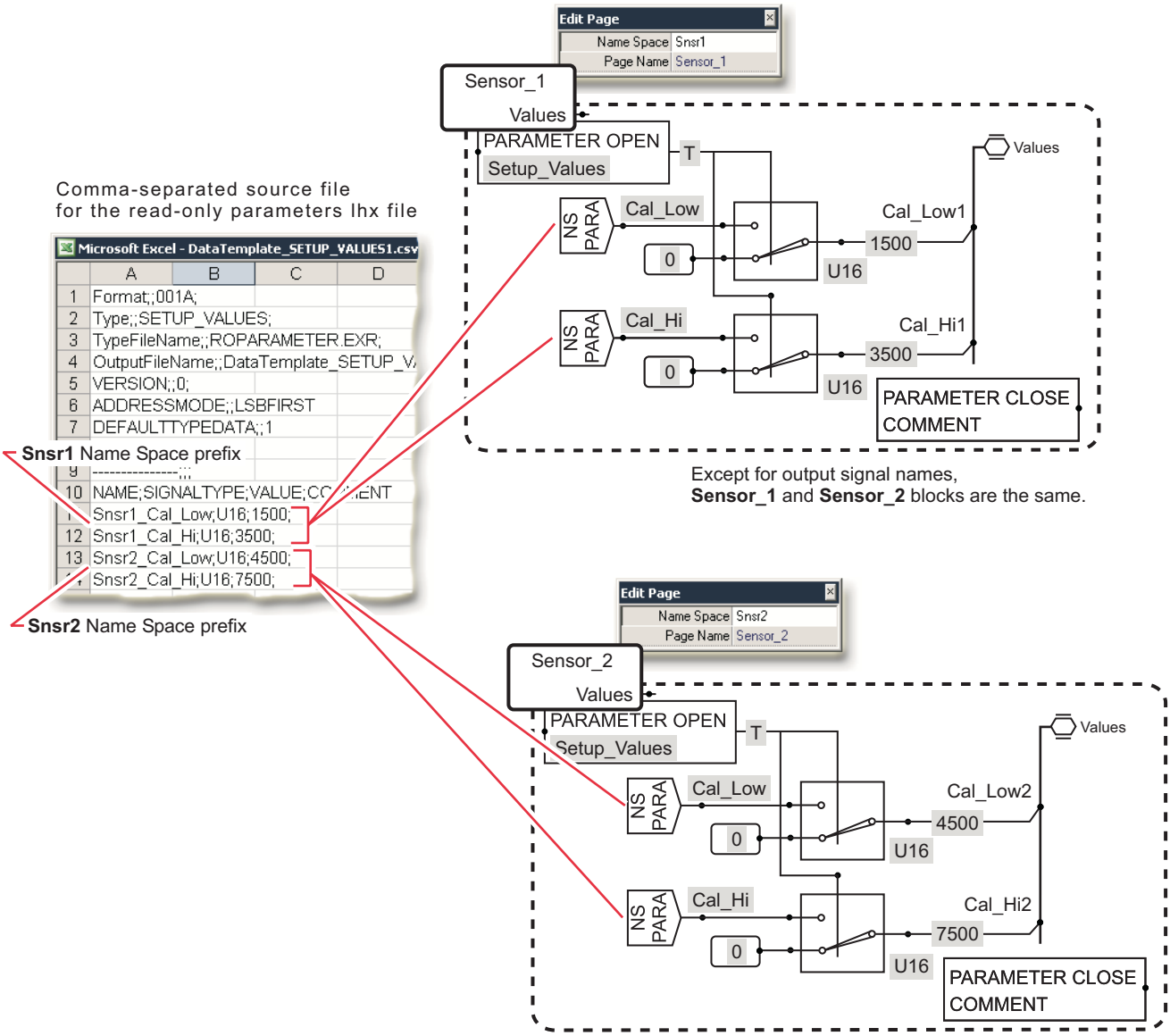
Names—OK
Data types—OK
Order—Not OK



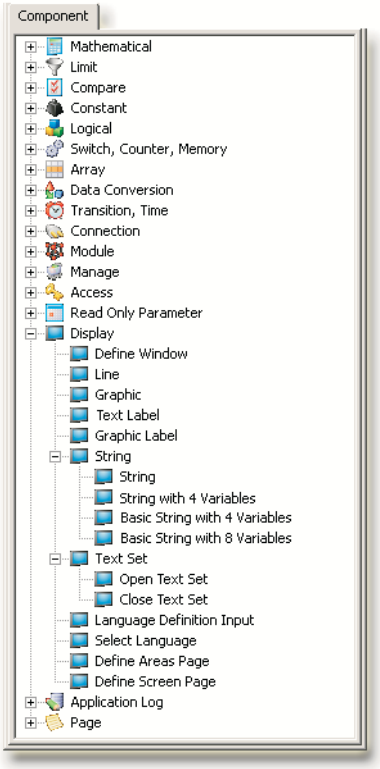
例: Read-only Parameter Input with Namespace

この例では以下の様子を示しています。

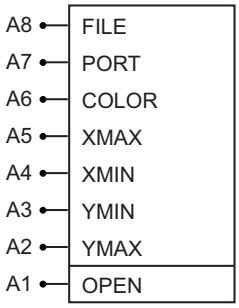
- 読み取り専用パラメータ lhx ファイルを Microsoft Excel で開いたところ
- ソースファイル内のそれぞれのパラメータ名が Name Space プレフィックスを持つところ



Display（ディスプレイ）



Define Window



用途

- 図形要素が表示されるウインドウを定義します。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートしています。スクリーンエディタでは、本コンポーネントは使用しません。

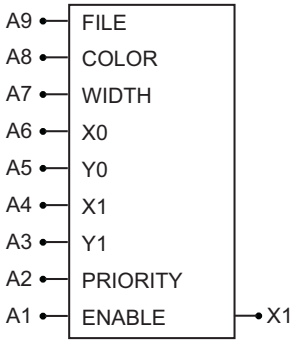
機能

- ハードウェアポートにファイルを割り当てます
- 以下の場合には
  - A1 = 0 の場合、ウインドウが表示されます
  - A1 = 1 の場合、ウインドウが表示されません
- A2～A5 によって、ピクセル単位の最小/最大ウインドウサイズが定義されます
  - A2 = y 軸上の最大ウインドウサイズ
  - A3 = y 軸上の最小ウインドウサイズ
  - A4 = x 軸上の最大ウインドウサイズ
  - A5 = x 軸上の最小ウインドウサイズ
- A6 = ウインドウの色
- A7 = ウインドウを出力するハードウェアポート
- A8 = 割り当てられたファイル名

有効接続

ピン	Type	ピン	Type
A1	BOOL	—	—
A2	INT		
A3–A5	INT		
A6	COL		
A7	PORT		
A8	FILE		

Line



用途

- IF_ASSIGN により定義されているウインドウに線を表示します。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートしています。スクリーンエディタでは、本コンポーネントは使用しません。

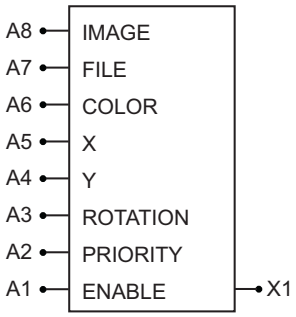
機能

- 以下の場合には
  - A1 = 0 の場合、X1 = 0 で線は表示されません
  - A1 = 1 の場合、X1 = 1 で線を表示することができます
- A2 = 線の表示優先度
- A3～A6 によって、IF_ASSIGN により定義されているウインドウ内に線が位置付けられます
  - A3 = ピクセル単位の線の Y 軸開始位置
  - A4 = ピクセル単位の線の X 軸開始位置
  - A5 = ピクセル単位の線の Y 軸終了位置
  - A6 = ピクセル単位の線の X 軸終了位置
- A7 = ピクセル単位の線幅、A3～A6 により定義されている点の間に線の中心
- A8 = 線の色
- A9 = IF_ASSIGN に割り当てられたファイル名

有効接続

ピン	Type	ピン	Type
A1	BOOL	X1	BOOL
A2	UINT		
A3–A6	INT		
A7	UINT		
A8	COL		
A9	FILE		

Graphic



用途

- IF_ASSIGN により定義されているウインドウにイメージファイル（***.bmp 等）を表示します。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートしています。スクリーンエディタでは、本コンポーネントは使用しません。

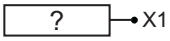
機能

- 以下の場合には
  - A1 = 0 の場合、X1 = 0 でイメージファイルは表示されません
  - A1 = 1 の場合、X1 = 1 でイメージファイルを表示することができます
- A2 = イメージファイルの表示優先度
- A3 = イメージファイルが回転する左回りの度数
- A4–A5 によって、IF_ASSIGN により定義されているウインドウ内にイメージファイルが位置付けられます
  - A4 = ピクセル単位のイメージファイルの Y 軸位置
  - A5 = ピクセル単位のイメージファイルの X 軸位置
- A6 = イメージファイルの背景色、カラーディスプレイに表示される白黒イメージファイルで使用
- A7 = IF_ASSIGN に割り当てられたファイル名
- A8 = イメージレジスタエディタ内 のイメージファイルに与えられたファイル名

有効接続

ピン	Type	ピン	Type
A1	BOOL	X1	BOOL
A2	UINT		
A3–A5	INT		
A6	COL		
A7	FILE		
A8	IMG		

Text Label



用途

- 外部 TL データ型信号を参照します。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートしています。スクリーンエディタでは、本コンポーネントは使用しません。

機能

- X1 = ?がユーザ定義名の場所です

有効接続

ピン	Type	ピン	Type
—	—	X1	TEXT

Graphic Label

用途

- 外部 PXO データ型信号を参照します。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートしています。スクリーンエディタでは、本コンポーネントは使用しません。

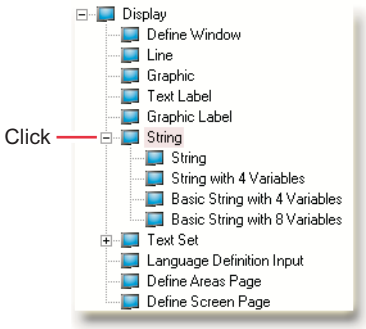
機能

- X1 = ?がユーザ定義名の場所です

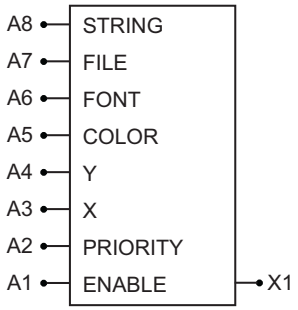
有効接続

ピン	Type	ピン	Type
—	—	X1	PXO

String


Display
Define Window
Line
Graphic
Text Label
Graphic Label
Click
String
String
String with 4 Variables
Basic String with 4 Variables
Basic String with 8 Variables
Text Set
Language Definition Input
Define Areas Page
Define Screen Page

String



用途

- IF_ASSIGN により定義されているウインドウにデータ変数を持たない単一文字列を表示します。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートするレガシーコンポーネントです。スクリーンエディタでは、本コンポーネントは使用しません。

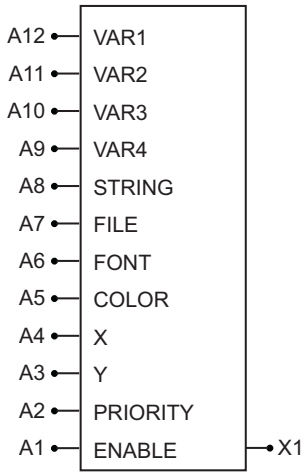
機能

- 以下の場合には
  - A1 = 0 の場合、X1 = 0 で文字列を表示できません
  - A1 = 1 の場合、X1 = 1 で文字列を表示できます
- A2 = 文字列の表示優先度
- A3-A4 によって、IF_ASSIGN により定義されているウインドウ内に文字列が位置付けられます
  - A3 = ピクセル単位の文字列の Y 軸開始位置
  - A4 = ピクセル単位の文字列の X 軸開始位置
- A5 = 文字列フォントの色
- A6 = 文字列フォントのタイプ
- A7 = IF_ASSIGN に割り当てられたファイル名
- A8 = テキストエディタを用いて割り当てられた文字列用の文字列ポインタ

有効接続

ピン	Type	ピン	Type
A1	BOOL	X1	BOOL
A2	FILE		
A3	FONT		
A4	COL		
A5	INT		
A6	INT		
A7	UINT		
A8	BOOL		

String with 4 Variables



用途

- IF_ASSIGN により定義されているウインドウに4つのデータ変数値を持つ文字列を表示します。
- 文字列位置、色、フォントを指定する必要がない場合には、IF_STRING-OUT を用いることができます。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートするレガシーコンポーネントです。スクリーンエディタでは、本コンポーネントは使用しません。

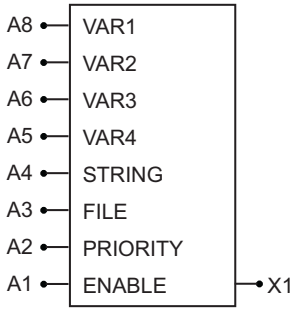
機能

- 以下の場合には
  - A1 = 0 の場合、X1 = 0 で文字列を表示できません
  - A1 = 1 の場合、X1 = 1 で文字列を表示できます
- A2 = 文字列の表示優先度
- A3-A4 によって、IF_ASSIGN により定義されているウインドウ内に文字列が位置付けられます
  - A3 = ピクセル単位の文字列の Y 軸開始位置
  - A4 = ピクセル単位の文字列の X 軸開始位置
- A5 = 文字列フォントの色
- A6 = 文字列フォントのタイプ
- A7 = IF_ASSIGN に割り当てられたファイル名
- A8 = テキストエディタを用いて割り当てられた文字列用の文字列ポインタ
- A9-A12 = テキストエディタを用いて割り当てられた文字列変数

有効接続

ピン	Type	ピン	Type
A1	BOOL	X1	BOOL
A2	UINT		
A3–A4	INT		
A5	COL		
A6	FONT		
A7	FILE		
A8	TEXT		
A9–A12	ALL		

Basic String with 4 Variables



用途

- IF_ASSIGN により定義されているウインドウに4つのデータ変数値を持つ文字列を表示します。
- 文字列位置、色、フォントを指定する必要がない場合には、IF_STRING-OUT を用いることができます。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートするレガシーコンポーネントです。スクリーンエディタでは、本コンポーネントは使用しません。

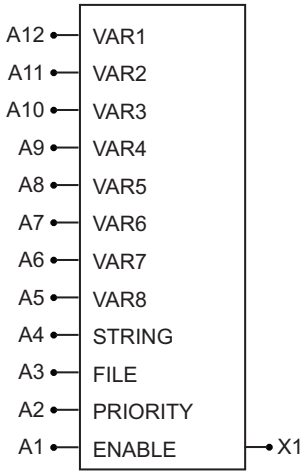
機能

- 以下の場合には
  - A1 = 0 の場合、X1 = 0 で文字列を表示できません
  - A1 = 1 の場合、X1 = 1 で文字列を表示できます
- A2 = 文字列の表示優先度
- A3 = IF_ASSIGN に割り当てられたファイル名
- A4 = テキストエディタを用いて割り当てられた文字列用の文字列ポインタ
- A5-A8 = テキストエディタを用いて割り当てられた文字列変数

有効接続

ピン	Type	ピン	Type
A1	BOOL	X1	BOOL
A2	UINT		
A3	FILE		
A4	TEXT		
A5–A8	ALL		

Basic String with 8 Variables



用途

- IF_ASSIGN により定義されているウインドウに8つのデータ変数値を持つ文字列を表示します。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートするレガシーコンポーネントです。スクリーンエディタでは、本コンポーネントは使用しません。

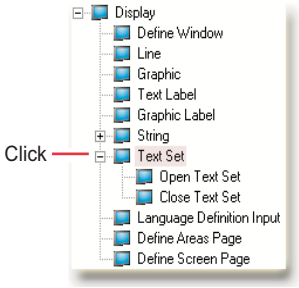
機能

- 以下の場合には
  - A1 = 0 の場合、X1 = 0 で文字列を表示できません
  - A1 = 1 の場合、X1 = 1 で文字列を表示できます
- A2 = 文字列の表示優先度
- A3 = IF_ASSIGN に割り当てられたファイル名
- A4 = テキストエディタを用いて割り当てられた文字列用の文字列ポインタ
- A5–A12 = テキストエディタを用いて割り当てられた文字列変数

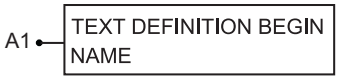
有効接続

ピン	Type	ピン	Type
A1	BOOL	X1	BOOL
A2	UINT		
A3	FILE		
A4	TEXT		
A5–A8	ALL		

Text Set



Open Text Set



用途

- 多言語フォーマット定義を含んだ外部ファイルをアプリケーションにアタッチします。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートするレガシーコンポーネントです。スクリーンエディタでは、本コンポーネントは使用しません。

機能

- A1 = ユーザのアプリケーションで使用する言語を定義します
- S1 = テキスト定義ファイルの別名

Close Text Set

TEXT DEFINITION END
COMMENT

用途

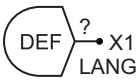
- テキスト定義機能を終了します。

本コンポーネントは、スクリーンエディタが利用できるようになる前に作成されたディスプレイアプリケーションをサポートするレガシーコンポーネントです。スクリーンエディタでは、本コンポーネントは使用しません。

機能

- コンポーネントと共に使用します

Language Definition Input（言語定義入力）



用途

- **Select Language** コンポーネントと組み合わせることによって、PLUS+1 ディスプレイに表示される言語を切り替えることができます。
  - *362*ページ *例: Select Language*に、**Select Language**コンポーネントと共に本コンポーネントを使用する方法が示されています。
  - 本コンポーネントをスクリーンエディタと共に使用する方法の詳細に関しては、*303* ページの「*Define Screen ページ/言語の切換え*」を参照してください。

機能

- ? = スクリーンエディタの **Text Groups**（テキストグループ）タブに定義されている言語

有効接続

ピン	データ型	ピン	データ型
—	—	X1	LANG

Select Language（言語選択）

A1 SELECT LANGUAGE

用途

- **Language Definition Input** コンポーネンと組み合わせることによって、PLUS+1 ディスプレイに表示される言語を切り替えることができます。
  - 以下の例に、**Language Definition Input** コンポーネントと共に本コンポーネントを使用する方法が示されています。
  - 本コンポーネントをスクリーンエディタと共に使用する方法の詳細に関しては、*303* ページの「*Define Screen ページ/言語の切換え*」を参照してください。

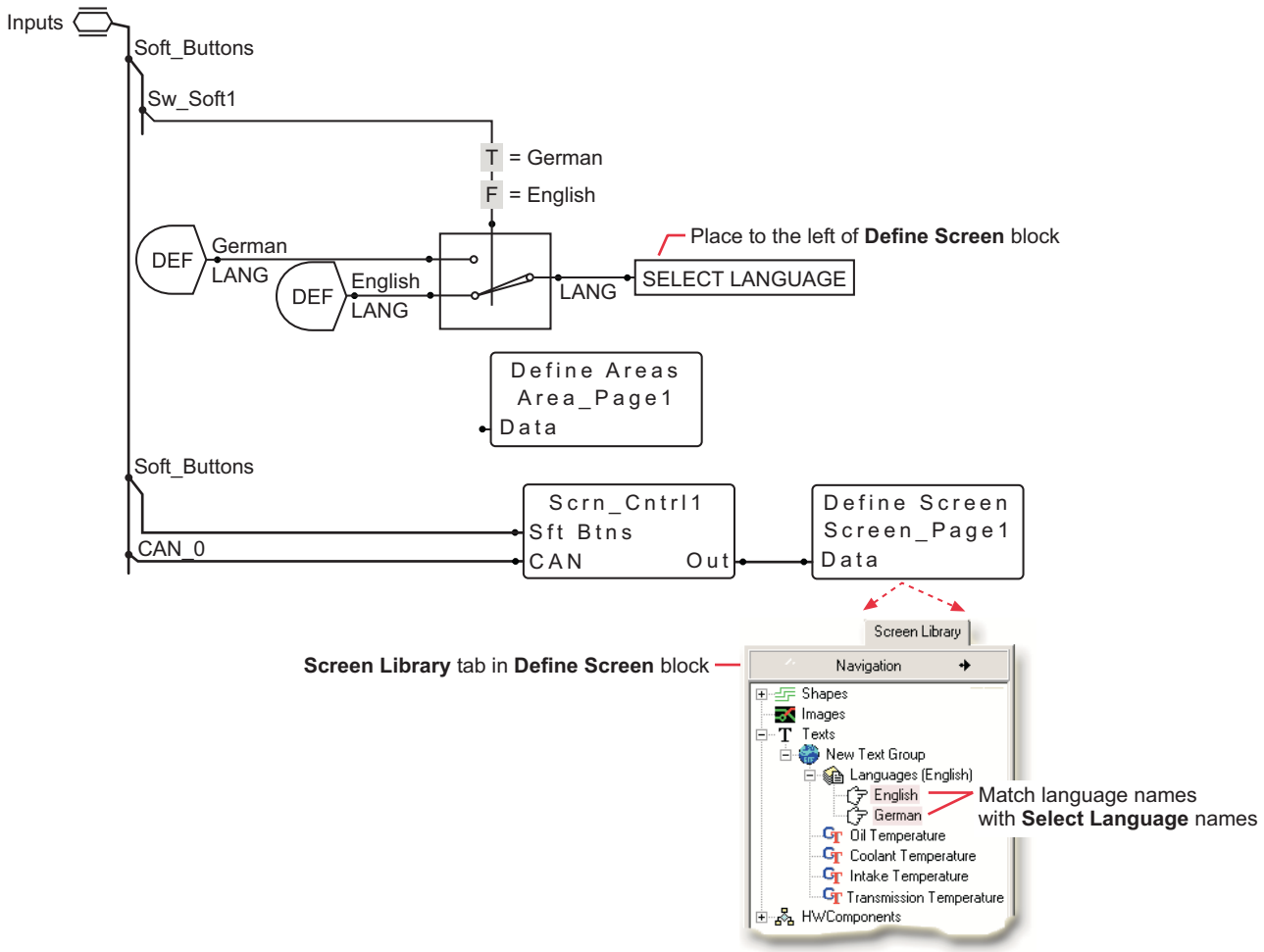
機能

- A1= **Define Screen** ブロックの **Screen Library** タブ定義されている言語
- **Define Screen** ブロックまたは **Define Applog** ブロックより左側に置く必要があります

有効接続

ピン	データ型	ピン	データ型
A1	LANG	—	—

例: Select Language/Language Definition



Define Areas Page（エリア定義ページ）

```
Define Areas
 Area_Page1
Data
```

PLUS+1 ディスプレイ（DP600 等）用のハードウェアファイルを **Project Manager** タブにインストールすると、このハードウェア依存コンポーネントを使用できるようになります。

File メニューの **Import Page** コマンドと **Import Block** コマンドはこのコンポーネントでは動作しません。**Define Screen** ブロックをインポートするとこれらの内容が欠落します。

用途

- PLUS+1 ディスプレイに表示されるエリアを定義します。

  本ページの使用の詳細に関しては、*307* ページから始まる*スクリーンエディタ*の章を参照してください。

Define Screen Page（スクリーン定義ページ）

```
Define Screen
Screen_Page1
Data
```

PLUS+1 ディスプレイ（DP600 等）用のハードウェアファイルを **Project Manager** タブにインストールすると、このハードウェア依存コンポーネントを使用できるようになります。

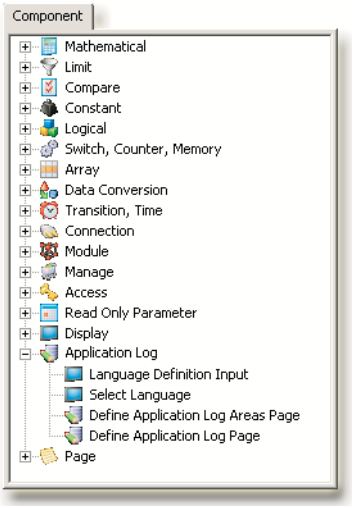
File メニューの **Import Page** コマンドと **Import Block** コマンドはこのコンポーネントには使えません。**Define Screen** ブロックをインポートするとこれらの内容が欠落します。

用途

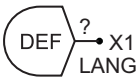
- PLUS+1 ディスプレイ画面エリアに表示される内容を定義します。

  本ページの使用の詳細に関しては、*307* ページから始まる*スクリーンエディタ*の章を参照してください。

Application Log（アプリケーションログ）



Language Definition Input（言語定義入力）



用途

- **Select Language** コンポーネントと共に使用し、アプリケーションデータログに異なる言語でデータを書き込むことができるようにします。
  - *366*ページの *例: Language Definition/Select Language*でこのコンポーネントをどのように**Select Language**コンポーネントと共に使用するかを示しています。

機能

- ? = **Define APPLog** ブロックの **Screen Library** タブ内で定義されている言語

有効接続

ピン	データ型	ピン	データ型
—	—	X1	LANG

Select Language

A1 SELECT LANGUAGE

用途

- **Lang** コンポーネントと共に使用し、アプリケーションデータログに異なる言語でデータを書き込むことができるようにします。
  - *366*ページの *例: Language Definition/Select Language*でこのコンポーネントをどのように**Language Definition Input**コンポーネントと共に使用するかを示しています。

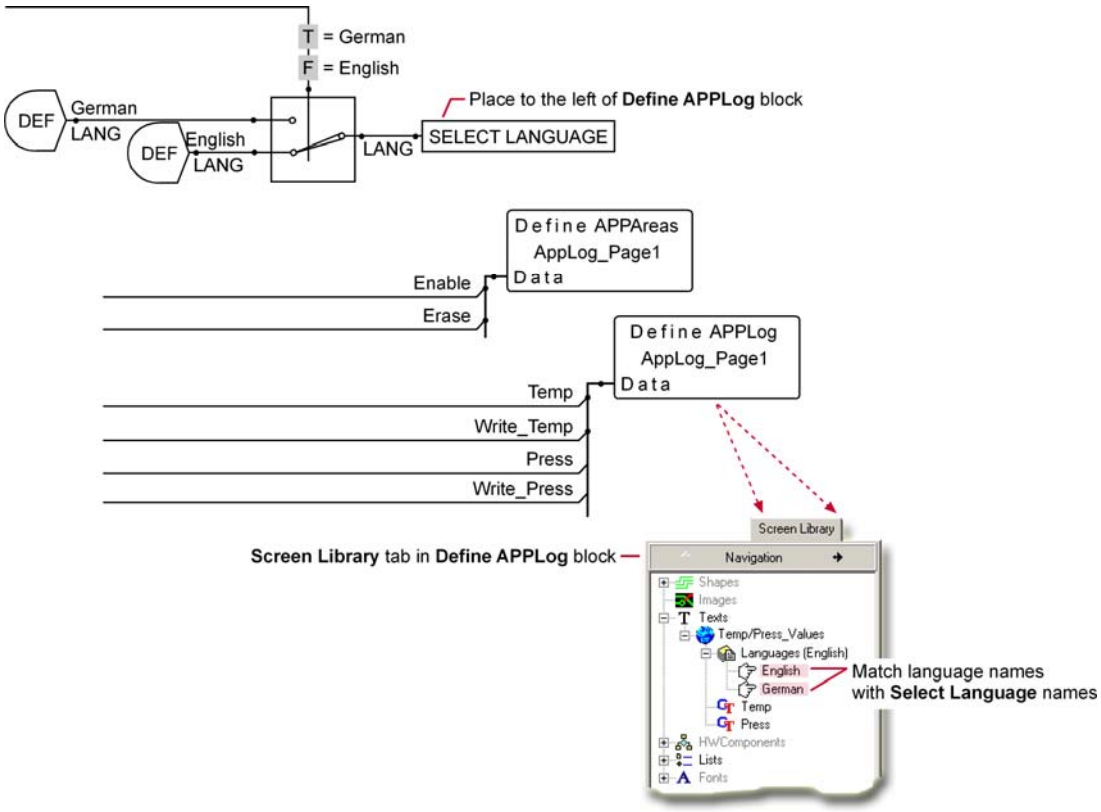
機能

- A1 = **efine APPLog** ブロックの **Screen Editor** タブで定義されている言語
- **Define Screen** ブロックまたは **Define Applog** ブロックより左側に置く必要があります。

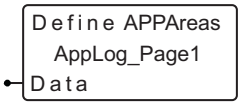
有効接続

ピン	データ型	ピン	データ型
A1	LANG	—	—

例: Language Definition/Select Language



Define Application Log Areas Page



用途

- **Define APPLog** コンポーネントと共に使用し、アプリケーションデータログ機能を実装します。

  このコンポーネントの使用に関する詳細な情報は *307* ページから始まる*スクリーンエディタ*の章を参照してください。

Define Application Log Page

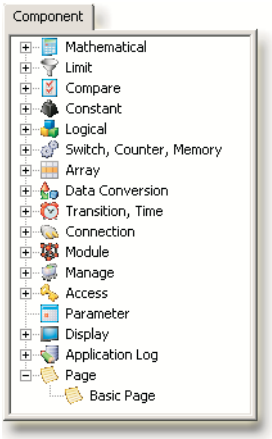
用途

- **Define APPAreas** コンポーネントと共に使用し、アプリケーションデータログ機能を実装します。

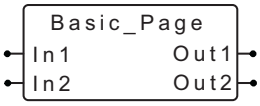
  このコンポーネントの使用に関する詳細な情報は *307* ページから始まる*スクリーンエディタ*の章を参照してください。

File メニューの **Import Page** コマンドと **Import Block** コマンドはこのコンポーネントには使えません。**Define APPLog** ブロックをインポートするとこれらの内容が欠落します。

Page（ページ）



Basic Page（基本ページ）



用途

- 独自のページの作成を開始するときに使える汎用ページです。

スクリーンエディタ

本章では PLUS+1 スクリーンエディタについて説明します。PLUS+1 GUIDE バージョン 2.0.3 以降はスクリーンエディタ付きです。

スクリーンエディタで DP600 などザウアーダンフォス社製 PLUS+1 ディスプレイ用のアプリケーション・ソフトウェアを作成できます。

本章で説明した項目を使って PLUS+1 GUIDE ディスプレイ用のアプリケーションを作成する方法については、*361* ページの「*スクリーンエディタの使い方*」を参照してください。

この章の内容

スクリーンエディタ 369
スクリーンエディタについて 370
スクリーンエディタのエレメント 371
Define Areas Page 373
Define Areas Page/Inspector Tab 375
Define Screen Page 382
Define Screen Page/Add Library Items 384
Define Screen Page/Inspector Tab 385
Define Screen Page/Image Register 391
Define Screen Page/Text Register 394

スクリーンエディタについて

スクリーンエディタは以下の構成になっています。

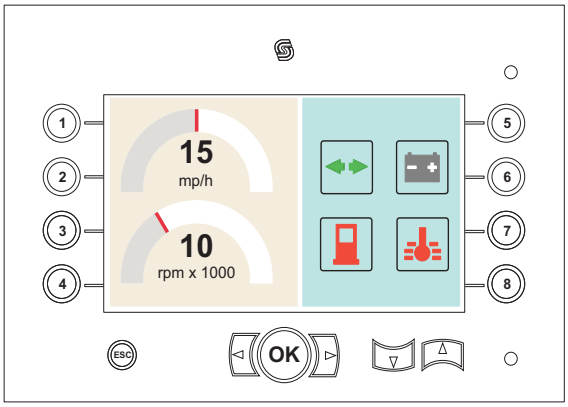
- **Define Areas Page**（エリア定義ページ）
- **Define Screen Page**（画面定義ページ）
- **Basic Page**（基本ページ）

これらのページで以下を定義します。

- ディスプレイに表示するスクリーン･エリア。
- 各スクリーン･エリアの内容（画像やテキストなど）。
- 個々の画面エレメント表示を制御する信号（これらのページに信号を引き込み後）。

Define Areas ページ、**Define Screen** ページ、**Basic Page** は PLUS+1 GUIDE バージョン 2.0.3 以降の Components タブにあります。グラフィックアプリケーションを作成するときは、これらのページをディスプレイテンプレートの **Application** ページにドラッグします。

一般的なディスプレイ



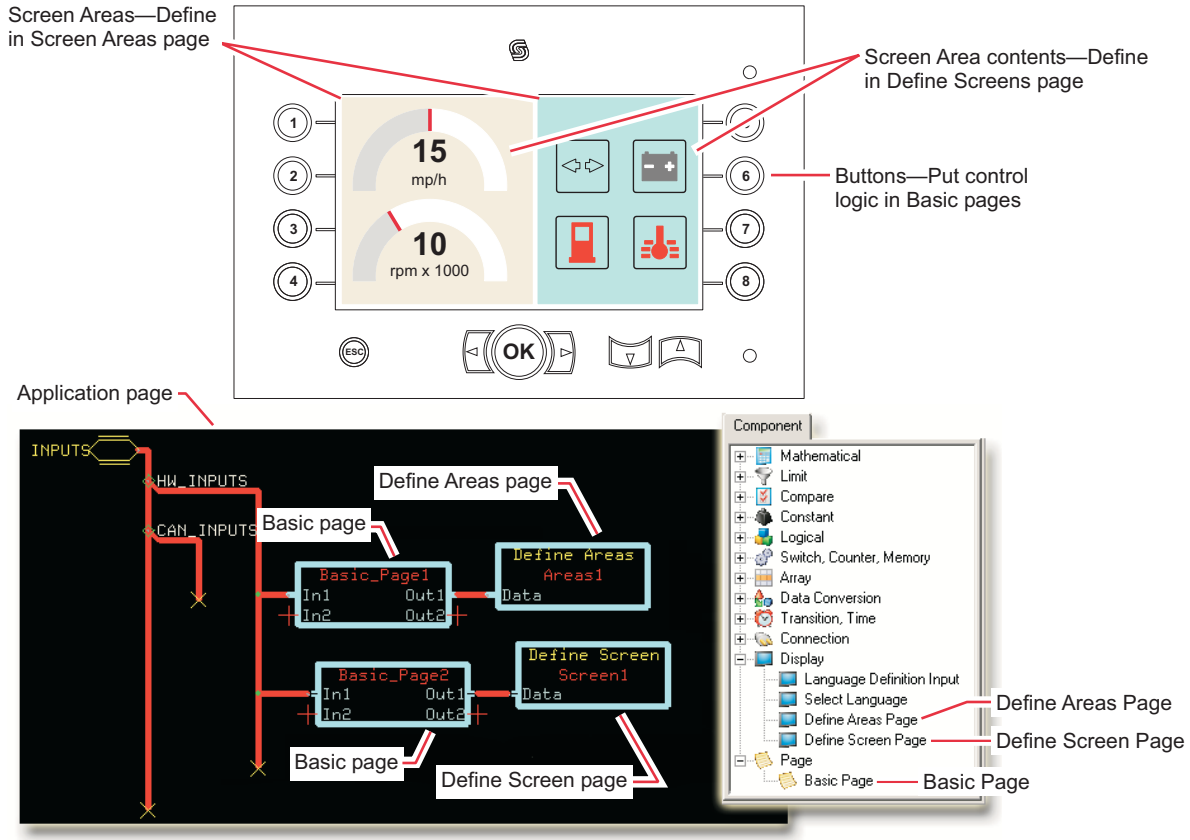
上図は一般的なディスプレイです。

スクリーンエディタで以下を設定しています。

- ベージュ色と水色のスクリーン･エリア。
- それぞれのスクリーン･エリアに表示するビットマップ画像とテキスト。
- 各スクリーン･エリアとその内容表示を制御するボタンや外部信号源からの信号。

スクリーンエディタではビットマップ画像を操作（回転、オンオフ）することはできますが、画像を新規作成したりグラフィック画面に合わせてサイズを調整したりすることはできません。ビットマップ画像を新規作成するには、PaintShop™ Pro®や Adobe Photoshop® Elements などの画素編集ソフトが必要です。

スクリーンエディタのエレメント



本図では以下が説明されています。

- ディスプレイ画面の基本エレメント。
- エレメントの定義と制御を行う **Define Areas** ページ、**Define Screen** ページ、**Basic Page**。これら 3 つのページはディスプレイテンプレートの **Application** ページに配置します。

ディスプレイテンプレートのインストール方法、テンプレートの **Application** ページへの上記ページの追加方法については、*333 ページの「PLUS+1 の使い方」*で説明します。

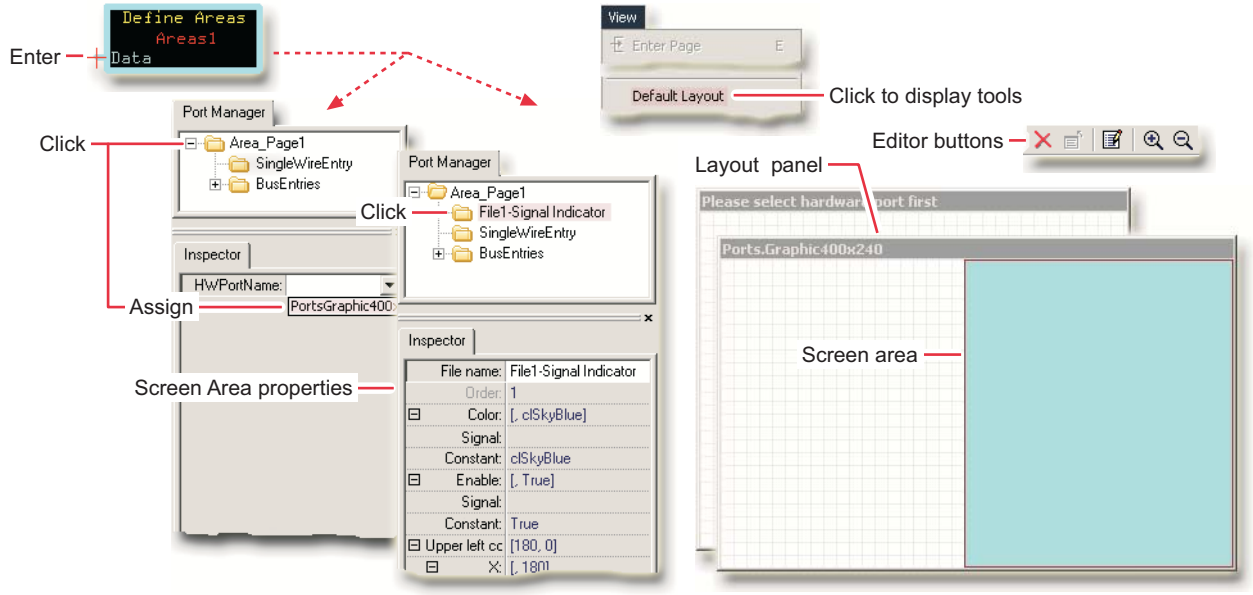
スクリーンエディタのエレメント

項目	説明
Application Page （アプリケーションページ）	ディスプレイ用アプリケーションの大部分はこのページで作成します。 テンプレートはアプリケーションを作成するディスプレイに合ったものを使用してください。 ディスプレイのアプリケーションは **Define Screen** ページ、**Define Areas** ページ、**Basic Page** を用いて作成します。**Component** タブでページを選択して、**Application** ページにドラッグします。 ディスプレイのボタンその他からの信号は、入力バスでこれらのページに送られます。

スクリーンエディタのエレメント

項目	説明
Define Areas Page （エリア定義ページ）	このページに入ったときに示されるツールを用いて、ディスプレイ画面に表示するスクリーン･エリアを作成します。 スクリーン･エリアとは画像、線、テキストが表示される背景のことです。 **Define Areas** ページのツールを用いてスクリーン･エリアの位置、大きさ、色を設定します。 スクリーン･エリアはディスプレイ画面の全体を占めても、一部だけを占めてもかまいません。 **Define Areas** ページに信号を送ってから、ページ内の編集ツールでスクリーン･エリアの表示を有効にする信号を選択します。 **Define Areas** ページはサブバスをサポートしていません。このページに信号を送るときは、必ずメインバスを使用してください。 **File** メニューの **Import** コマンドはこのコンポーネントでは使用できません。このコンポーネントを新しいプロジェクトにインポートするとその内容が欠落します。
Define Screen Page （画面定義ページ）	このページに入ったときに示されるツールを用いて、各スクリーン･エリアの内容を設定します。 スクリーン･エリアの内容とはビットマップ画像、線、テキストなどのことです。 （Screen Area の内容を定義する前に、まず **Define Areas** ページを使って Screen Area を作成する必要があります。） **Define Screen** ページのツールを用いてスクリーン･エリアに表示する各項目の外観と位置を設定します。 **Define Screen** ページに信号を送ってから、このページの編集ツールでスクリーン･エリアの項目の動作を制御する信号を選択します。 信号を用いて項目の表示と非表示、移動と回転、変動数値の表示を行います。 **Define Screen Page** サブバスをサポートしていません。このページに信号を送るときは、必ずメインバスを使用してください。 **File** メニューの **Import** コマンドはこのコンポーネントでは使用できません。このコンポーネントを新しいプロジェクトにインポートするとその内容が欠落します。
Basic Page（基本ページ）	このページはスクリーン･エリアの表示と項目を制御するロジックを入れておくために使ってください。 このページはプログラミングの便宜のために設けています。入出力ポート以外は何もないページです。

Define Areas ページ(エリア定義ページ)



前出の図はスクリーン･エリアを作成するときに用いる **Define Areas Page** のツール(タブ、ペイン、ボタン)を示しています。**Define Areas Page** に入ると、これらのボタンがあります。

(**Screen Library** タブもありますが、スクリーン･エリアの作成には使用しません。ここでは、このタブは省きます。)

Define Areas ページはサブバスをサポートしていません。このページに信号を送るときは、必ずメインバスを使用してください。

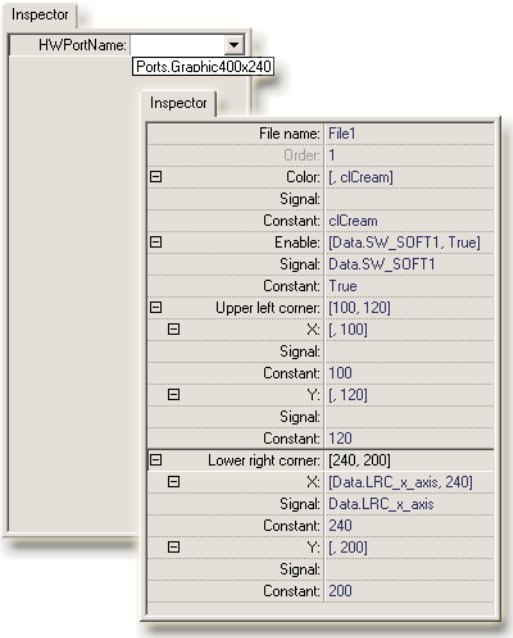
Screen Area を作成する前に、**Inspector** タブの **HWPortName** の割当をしてください。

File メニューの **Import Page** コマンドと **Import Block** コマンドは **Define Areas** ブロックでは使用できません。**Define Areas** ブロックを新しいプロジェクトにインポートするとその内容が欠落します。

Define Areas ページ

項目	説明	
Port Manager タブ	スクリーン･エリアの管理に使用します。 **Areas**：クリックするとハードウェアポートに割当てられたすべてのスクリーン･エリアをツリー表示します。**Inspector** タブの **HWPortName** は全スクリーン･エリアのポート割当を示します。 スクリーン･エリア名をクリックすると、**Inspector** タブのプロパティを表示し、そのエリアをレイアウトペインの前面に移動します。 **SingleWireEntry**：ここでは使用しません。 **BusEntries**：クリックすると、**Define Areas** ページの **Data** バスポートの全信号をツリー表示します。	
Inspector タブ	スクリーン･エリアのプロパティ管理に使用します。	
Layout ペイン	スクリーン･エリアのレイアウトに使用します。ペイン内のスクリーン･エリアを必要なだけドラッグして、サイズと位置を変更します。 このペインでスクリーン･エリアをクリックすると、そのエリアを前面に表示して、**Port Manager** タブは名前をハイライト表示、**Inspector** タブはプロパティを表示します。 スクリーン･エリアを右クリックすると、**Delete**、**Move Backward**、**Move Forward** コマンドのポップアップメニューが開きます。 **Inspector** タブの **Order** プロパティを変更するときは、**Move Backward** と **Move Forward** コマンドを使用します。	
Editor ボタン	これらのボタンを表示する行は、モニターのサイズで決まります。	
	閉じる	クリックすると、**Application** ページに戻ります。
	オプション	クリックすると、**Options** ウィンドウが開きます。 このウィンドウで、レイアウトペインの格子とスクリーン･エリアにある項目の透明性を調節します。
	拡大/縮小	レイアウトペインを拡大/縮小します。

Define Areas ページ/Inspector タブ



このタブはスクリーン･エリアのプロパティ管理に使用します。

スクリーン･エリアのプロパティを表示するには、**Port Manager** タブに一覧表示されたスクリーン･エリア名をクリックするか、レイアウトペインのスクリーン･エリアをクリックします。

Define Areas ページ/Inspector タブ

項目	説明
HWPortName（ハードウェアポート名）	**Port Manager** タブの **Areas1** をクリックすると使用可能になります。 プルダウンリストから **HWPortName** をクリックして選択すると、**Area1** ツリーの全スクリーン･エリアのポートタイプと最大画素数が設定されます。 **HWPortName** を選択すると、**Inspector** タブの他のプロパティが使用可能になります。
File name（ファイル名）	スクリーン･エリア名
Order（順序）	複数のスクリーン･エリアの **Enable**（有効化）プロパティが同時に **True** となった場合（**Entry** 信号が True であるか、**Value** が **True** である場合）のスクリーン･エリアの表示順序を示します。 **Port Manager** タブでスクリーン･エリアを割当てる順序に従って、**Order** プロパティの初期値が設定されます。 **Order** プロパティが **1** のスクリーン･エリアは、**Order** プロパティが **2** のスクリーン･エリアの手前に表示され、**Order** プロパティが **2** のスクリーン･エリアは、**Order** プロパティが **3** のスクリーン･エリアの手前に表示されます。 **Order** プロパティを変更するときは、スクリーン･エリアを右クリックします。開いたポップアップメニューで、**Move Forward**（前面に移動）または **Move Backward**（背面に移動）コマンドをクリックします。 Signal によって有効化されたスクリーン･エリアは、Constant を True に設定して有効化されたスクリーン･エリアより常に手前に表示されます。
Color（色）	**Signal**： スクリーン･エリアの色を制御する信号を選択するときにクリックします。 **Constant**： 標準色のリストを参照するときにクリックします。ダブルクリックすると、Windows の標準パレットが開いて、色を選択することができます。

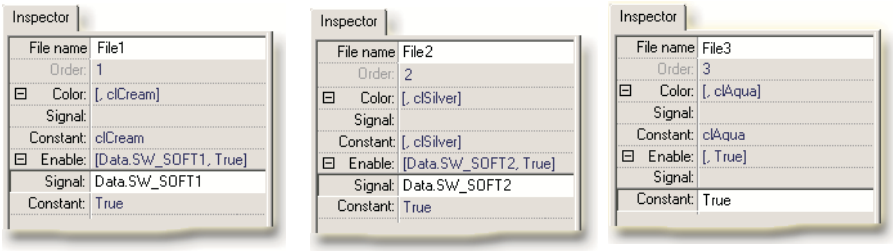
Define Areas ページ/Inspector タブ

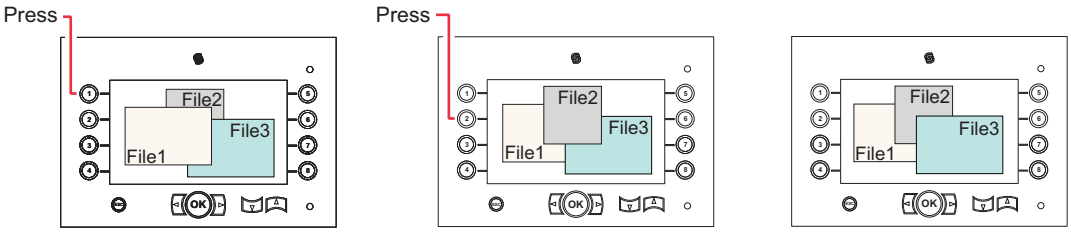
項目	説明
Enable（有効化）	スクリーン･エリアの表示を有効にします。 **Signal**： クリックしてプルダウンリストから信号を選択します。 True 信号によってこの画面はディスプレイペインの前面に出ます。 True 信号で **Color**、**Upper left Corner**、**Lower right corner** のプロパティが表示されます。 **Constant**：**True** または **False** をクリックします。 **True** に設定されたスクリーン･エリアは Signal で有効化されたスクリーン･エリアが True 信号を受取るまで前面に表示されます。 **False** を選択すると、スクリーン･エリアの表示が無効になります。この画面を選択することはできなくなります。
Upper left corner（左上隅）	スクリーン･エリアの左上隅の位置を設定します。 **Upper left corner**：スクリーン･エリアの左上隅の **X**（横）、**Y**（縦）画素位置を設定します。 **Signal**： 左上隅位置を制御する信号を選択するときにクリックします。 画素は無効にしても別のスクリーン･エリアの画素で上書きされるまでは表示されますから、スクリーン･エリアを縮小した効果を見たいときは背景となるスクリーン･エリアが必要です。 **Constant**：左上隅位置を固定値に設定するときにクリックします。 レイアウトペインでスクリーン･エリアをドラッグしても、これらの隅位置は変更されます。
Lower right corner（右下隅）	スクリーン･エリアの右下隅の位置を設定します。 **Lower right corner**： スクリーン･エリアの右下隅の **X**（横）、**Y**（縦）画素位置を設定します。 **Signal**： 右下隅位置を制御する信号を選択するときにクリックします。 **Constant**：右下隅位置を固定値に設定するときにクリックします。

Define Areas ページ / Enable プロパティについて

Inspector タブの **Enable** プロパティを選択して、スクリーン・エリアの表示プロパティを設定します。

- Signal 値により有効化されたスクリーン・エリア:
  - ディスプレイの電源を入れた後は表示されず、Enable の Signal 値が True になると表示されます。
  - Enable の Signal 値が True になると前面に移動します。
- Constant: True により有効化されたスクリーン・エリアは、Signal 値により有効化されたスクリーン・エリアが True の信号を受取るまでは常に前面に表示されます。

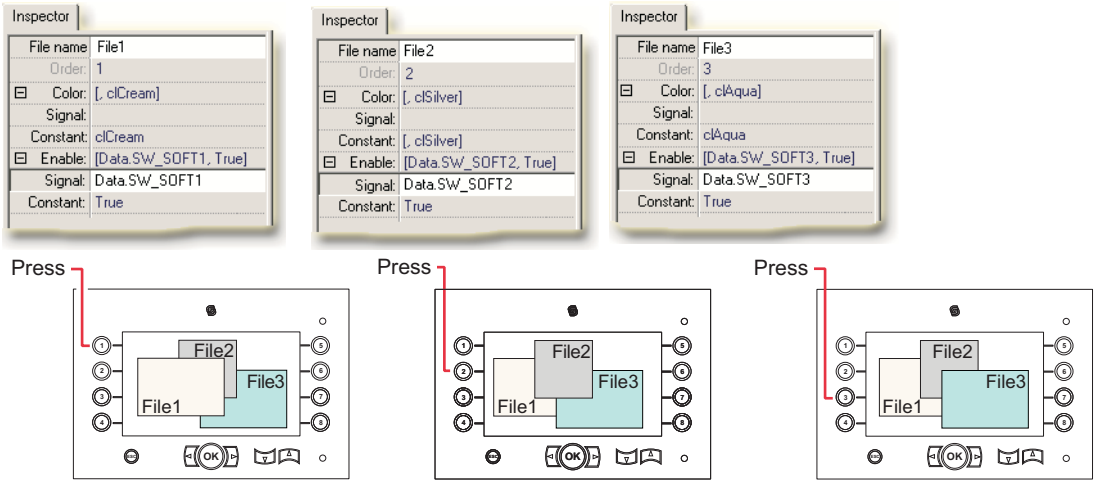




1. Data.SW_SOFT1 = True.
   Data.SW_SOFT2 = False.
   (File3 = Value of True.)
2. Data.SW_SOFT1 = False.
   Data.SW_SOFT2 = True.
3. Data.SW_SOFT1 = False.
   Data.SW_SOFT2 = False.

上図は True により有効化された 1 つのスクリーン・エリアと、Signal 値により有効化された 2 つのスクリーン・エリアの表示関係を示します。

各スクリーン・エリアの **Inspector** タブの **Enable** 設定に注意してください。



1. Data.SW_SOFT1 = True.
   Data.SW_SOFT2 = False.
   Data.SW_SOFT3 = False.

2. Data.SW_SOFT1 = False.
   Data.SW_SOFT2 = True.
   Data.SW_SOFT3 = False.

3. Data.SW_SOFT1 = False.
   Data.SW_SOFT2 = False.
   Data.SW_SOFT3 = True.

上図は Signal 値によって有効化された 3 つのスクリーン･エリアの表示関係を示します。

各スクリーン･エリアの **Inspector** タブの **Enable** 設定に注意してください。

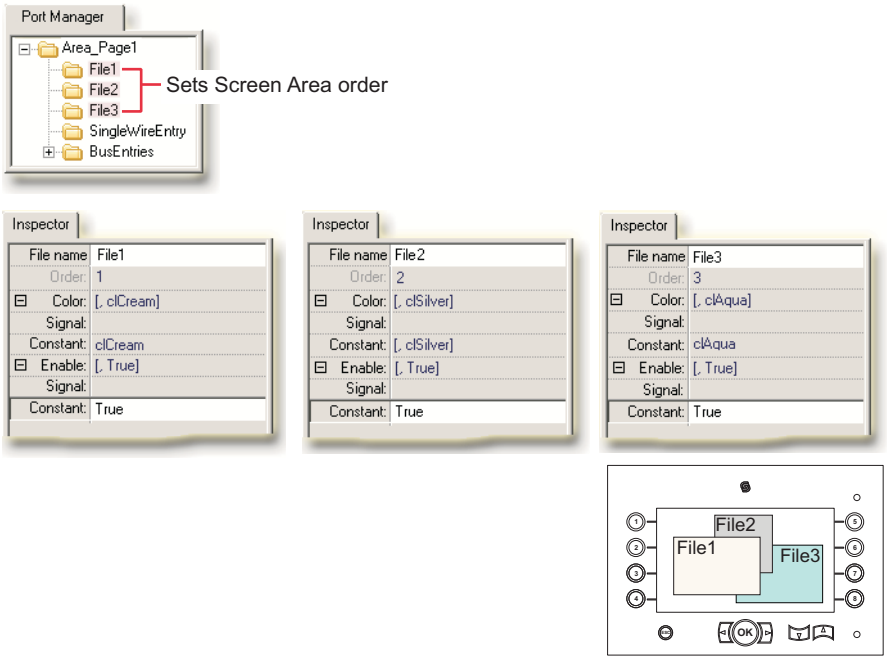
Define Areas ページ / Order プロパティについて

Inspector タブの **Order** プロパティは、複数のスクリーン･エリアの **Enable** が同時に以下の場合のときの前後の表示順序を設定します。

- **Signal** 値が **True**
- **Value** が **True**

（**Enable**プロパティの詳細については、***377***ページ*Define Areas ページ / Enable プロパティについて*を参照）

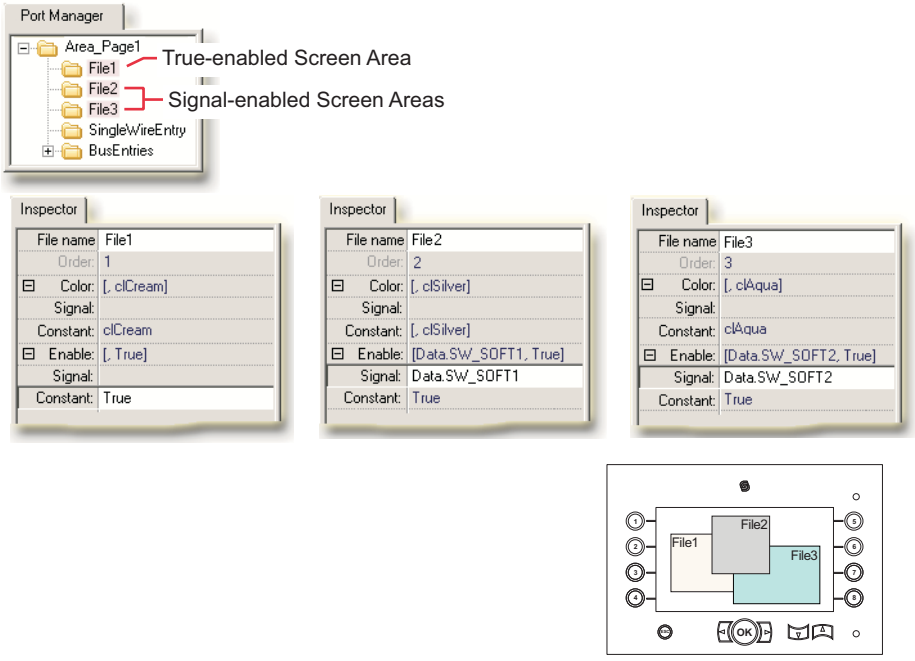
Port Manager タブでスクリーン･エリアを割当てる順序により、当初の前後 **Order** が設定されます。



File1, File2, and File3 = Value of True.

上図は True 値により有効化された 3 つのスクリーン･エリアの表示関係を示します。

Order プロパティを変更するときは、レイアウトペインでスクリーン･エリアを右クリックします。開いたポップアップメニューで、**Move Forward** または **Move Backward** コマンドをクリックします。



File1 = Value of True.
Data.SW_SOFT1 = True.
Data.SW_SOFT2 = True.

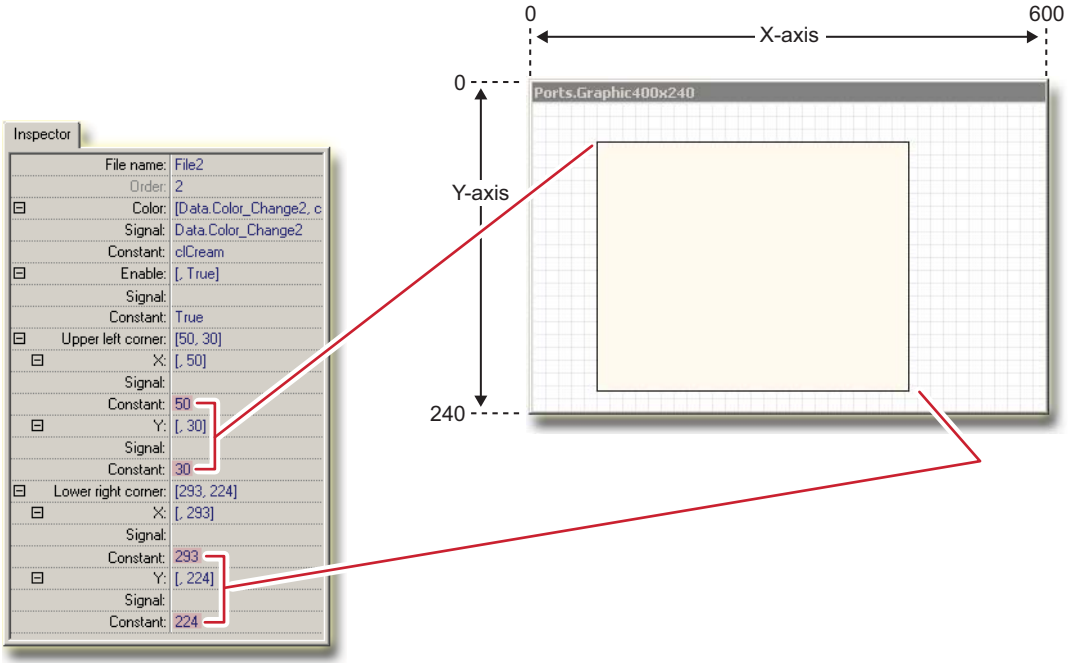
上図は以下の 3 つのスクリーン･エリアの表示順序を示します。

- **File1** のスクリーン･エリアは True により有効化されています。
- **File2** と **File3** のスクリーン･エリアは Signal 値により有効化されており、どちらの Signal 値も True です。

True の Signal 値によって有効化されたスクリーン･エリアは、次のように表示されます。

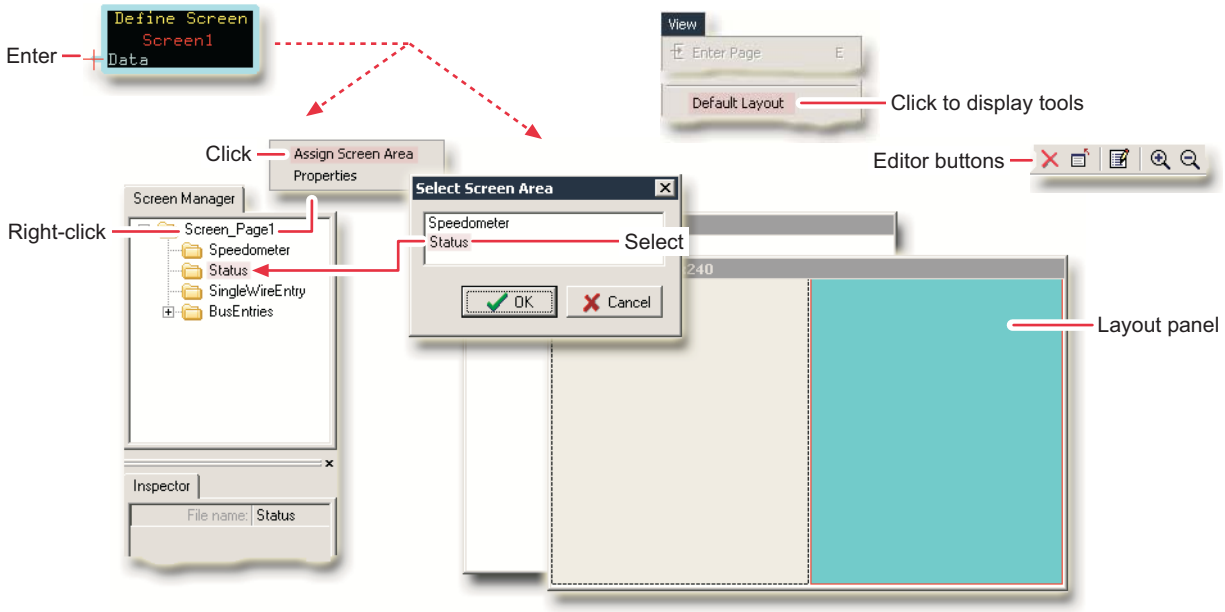
- True により有効化されたスクリーン･エリアの前。
- **Order** プロパティで設定された前後順。

Define Areas ページ / Corner プロパティについて



上図はスクリーン･エリアのサイズと位置に関して、**Inspector** タブの **Upper left corner** と **Lower right corner** プロパティによる効果を示します。

Define Screen ページ



初めて**Define Screen**ページに入ると、**Define Areas**ページ（***373***ページ）で新規作成したスクリーン･エリアを割当てるのに必要なツールが表示されています。

1 番目のスクリーン･エリアの割当を行うと、**Screen Library**タブ（上図には表示せず）からスクリーン･エリアに項目をドラッグできるツールが使用可能になります。***384***ページの「*Define Screen ページ / Library* アイテムの追加」で、これらのツールについて説明します。

Define Screen Page はサブバスをサポートしていません。このページに信号を送るときは、必ずメインバスを使用してください。

File メニューの **Import Page** コマンドと **Import Block** コマンドは **Define Screens** ブロックでは使用できません。**Define Screens** ブロックを新しいプロジェクトにインポートするとその内容が欠落します。

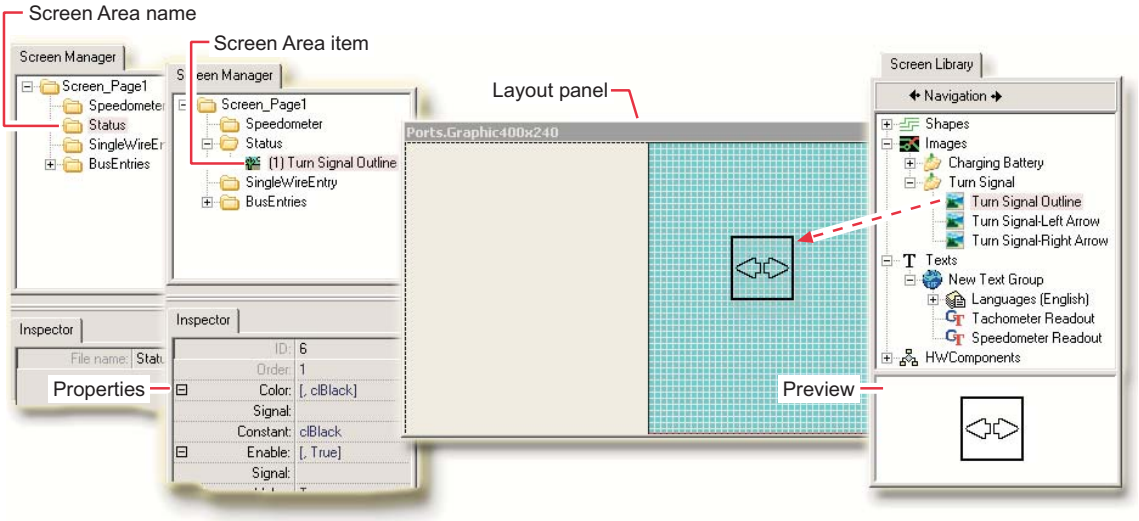
Define Screen ページ / スクリーン･エリア割当

項目	説明
Screen Manager タブ	このタブを使用して、**Define Areas** ページで作成したスクリーン･エリアをハードウェアポートに割当てます。 **Screen_Page 1**: クリックすると **Inspector** タブで選択した **PortName** に割当てたスクリーン･エリアをツリー表示します。 ツリーのスクリーン･エリア名をクリックすると、そのエリアをレイアウトペインの前面に移動します。 **SingleWireEntry**: ここでは使用しません。 **BusEntries**: クリックすると、**Define Screen** ページの **Data** ポートの全信号をツリー表示します。
Inspector タブ	**Screen1** ツリーのスクリーン･エリアをハードウェアポートに割当てます。
Layout ペイン	**Screen1** ツリーに追加したスクリーン･エリアを表示します。 （**Color**、**Enable**、**Corner** プロパティなどスクリーン･エリアのプロパティを変更したいときは、**Define Areas** ページに戻ってください。） スクリーン･エリアをクリックすると、そのエリアが前面に移動し、**Screen Manager** タブにあるそのエリア名がハイライト表示されます。

Define Screen ページ / スクリーン･エリア割当

項目		説明
Editor ボタン		これらのボタンを表示する行は、モニターのサイズで決まります。
	閉じる	クリックすると、**Application** ページに戻ります。
	プレビュー	クリックすると、レイアウトがディスプレイでどう見えるかプレビューできます。 プレビューは一時的に **Options** ウィンドウの設定に優先します。
	オプション	クリックすると、**Options** ウィンドウが開きます。 このウィンドウで、レイアウトペインの格子と透明性設定を調節します。
	拡大/縮小	レイアウトペインを拡大/縮小します。

Define Screen ページ / Library アイテムの追加



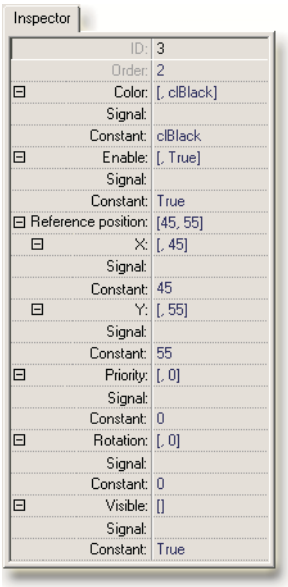
スクリーン･エリアをポートに割当てると、**Define Screen** ページにツールがさらに表示されます。

これらのツールを用いて、**Shapes**、**Images**、**Texts** など **Screen Library** タブの項目をスクリーン･エリアにドラッグして、これらの項目のプロパティを管理します。

Define Screen ページ/ライブラリ項目の追加

項目	説明
Screen Manager タブ	各スクリーン･エリアの内容管理に使用します。 スクリーン･エリア名をクリックすると、そのエリアに割当てるライブラリの全項目をツリー表示します。 このタブのスクリーン･エリア項目をクリックすると、**Inspector** タブにあるその項目のプロパティを表示します。 このツリーの項目を右クリックすると、**Cut**、**Copy**、**Delete**、**Paste** コマンドがあるポップアップメニューが開きます。
Inspector タブ	スクリーン･エリアの項目のプロパティ管理に使用します。
Layout ペイン	**Screen Library** タブからライブラリ項目をここのスクリーン･エリアにドラッグします。 スクリーン･エリアをクリックすると、そのエリアが前面に移動し、**Screen Manager** タブにあるそのエリア名がハイライト表示されます。 ライブラリ項目をクリックすると、**Inspector** タブにあるその項目のプロパティを表示します。**Screen Manager** タブは選択した項目名をハイライト表示します。
Screen Library タブ	スクリーン･エリアに追加できる **Shapes**、**Images**、**Texts**、**HWComponents** のライブラリをツリー表示します。 **Image Register** ペインを使用して、画像ライブラリに使用可能な **Images** を追加します。
Preview(プレビュー)	ビットマップ **Images** のプレビューを表示します。

Define Screen ページ / Inspector タブ



スクリーン･エリアに追加するライブラリ項目のプロパティを管理するタブです。

ライブラリ項目のプロパティを表示するには、**Screen Manager** タブの項目名をクリックするか、レイアウトペインでその項目をクリックします。

ライブラリ項目は種類ごとにプロパティが異なりますから、選択する項目の種類で、**Inspetor** タブに表示するプロパティが決まります。

Shapes、**Images**、**Texts**、**HWComponents**のプロパティリストは、***387***ページ *Define Screen ページ / Screen アイテムのプロパティについて* を参照してください。

Define Screen ページ/Inspector タブ

項目	説明
ID	内部識別に使用します。この値は変更できません。
Order（順序）	重なり合うライブラリ項目が同時に **Signal** 値が True または **Constant** が **True** で **Enable** になったときの前後の表示順序を示します。 **Order** が **1** の項目は、**Order** が **2** の項目の手前に表示します。 ライブラリ項目をレイアウトペインにドラッグする順序で、**Order** の初期値が設定されます。 ライブラリ項目の **Order** を変更するときは、項目を右クリックし、表示されるポップアップメニューで、**Move Forward** または **Move Backward** コマンドをクリックします。
Begin point（始点）	**X: Line** の X（横）軸始点を画素で設定します。 **Y**: **Line** の Y（縦）軸始点を画素で設定します。 （**Screen Library** タブの **Shapes** のツリーから **Line** を選択します。）
End point（終点）	**X**： **Line** の X（横）軸終点を画素で設定します。 **Y**： **Line** の Y(縦）軸終点を画素で設定します。
Width（幅）	X（横）軸に引いた **Line** の上下の厚さを画素で設定します。 Y（縦）軸に引いた **Line** の左右の厚さを画素で設定します。

Define Screen ページ/Inspector タブ

項目	説明
Color（色）	ライブラリ項目の色を設定します。 信号のリストから信号を選択して色を設定するときは、**Signal** をクリックします。 色リストから色を選択するときは、**Constant** をクリックします。Windows の標準カラーパレットから色を選択するときは、**Constant** をダブルクリックします。
Font（フォント）	**Screen Library** からとった **Texts** のフォントプロパティを設定します。
Enable（有効化）	ライブラリ項目の表示を有効にします。
Reference position（基準位置）	**Images** と **HWComponents** ライブラリ項目に表示する小さな青十字の **X** 軸座標、**Y** 軸座標を設定します。 ライブラリ項目の縦移動、横移動、回転はすべてこの点が基準になります。 **Image Register** ペインの **Reference point** でこの十字の位置を設定します。
Priority（優先順位）	複数のライブラリ項目が同時に True **Signal** で **Enable** のときに、どの項目を表示するかを決定します。 複数の項目が有効なとき、ディスプレイは **Priority** が最も高い項目だけを表示します。 **Priority** の数字が小さいほど、表示の優先順位は高くなります。**0** の項目は最優先表示順位です。
Rotation（回転）	**Images** や **HWComponents** ライブラリを回転させます。 全ての回転動作はこのライブラリ項目に表示された小さな青十字が基準になります。 **Image Register** ペインの **Reference point** でこの十字の位置を設定します。 回転の値は **0** から **359°**の範囲で設定します。
Visible（可視）	ライブラリ項目が可視になると、True のブール信号を出力します。 可視になるには、True のスクリーン･エリア、True の **Enable** プロパティ、そして同じスクリーン･エリアにある他のライブラリ項目の **Priority** と同等ないしそれ以上の **Priority** が必要です。 （この信号を出力するには、**Data** バスに結線してください。）

Define Screen ページ / Screen アイテムのプロパティについて

ライブラリ項目の種類によりプロパティは異なります。下表は **Shapes**、**Images**、**Texts**、**HWComponents** のプロパティを示します。

Properties of Screen Library tab items

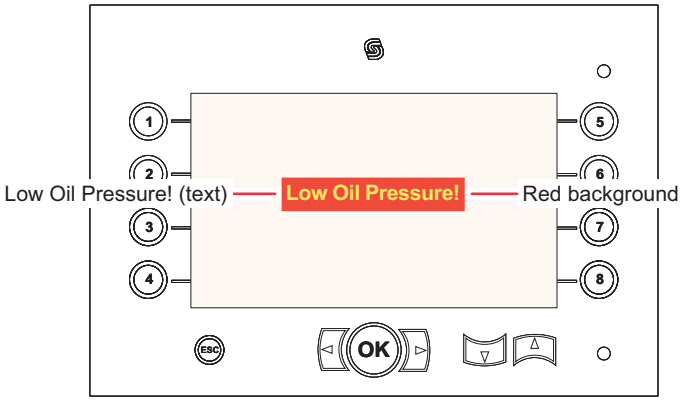
プロパティ	ライブラリ項目			
	形状	画像	テキスト	ハードウェアコンポーネント
Begin point（始点）	✔	—	—	—
Color（色）	✔	✔	✔	✔
Enable（有効化)	✔	✔	✔	✔
End point（終点）	✔	—	—	—
Font（フォント）	—	—	✔	—
ID	—	✔	✔	✔
Reference position（基準位置）	—	✔	—	✔
Order（順序）	✔	✔	✔	✔
Priority（優先順位）	✔	—	✔	✔
Rotation（回転）	—	✔	—	✔
Starting point (始点)	—	—	✔	—
Visible（可視）	✔	✔	✔	✔
Width（幅）	✔	—	—	—

Define Screen ページ / Order（順序）プロパティについて

Inspector タブには **Order** と **Enable** のプロパティがあり、その組み合わせにより表示が異なります。

以下の例では、

- **Low Oil Pressure!** のテキストは **Order** プロパティが **1** で、**Enable** プロパティが変化します。
- 赤の背景は **Order** プロパティが **2** で、**Enable** プロパティが変化します。



Order and Enable properties

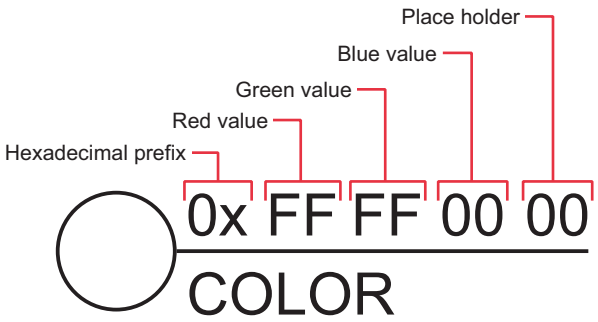
項目	Order	Enable	結果
「Low Oil Pressure!」文字	1	Constant	常時表示、警告テキストは背景バーの前面。
赤背景	2	Constant	
「Low Oil Pressure!」文字	1	Signal	背景バー常時表示。 警告テキストは信号が真のときだけ背景バーの前面に表示。
赤背景	2	Constant	
「Low Oil Pressure!」文字	1	Signal	警告テキストと背景バーは、それらの信号が真のときだけ表示。 両方の信号が真のときに警告テキストを背景バーの前面に表示。
赤背景	2	Signal	
「Low Oil Pressure!」文字	1	Constant	警告テキストを常時表示。 背景バーはその信号が真のときに警告テキストの後ろに表示。
赤背景	2	Signal	

Define Screen ページ / Color プロパティについて

Inspector タブには **Color** プロパティがあり、**Constant** または **Signal** によって色を設定できます。

Signal で色を設定するときは、**Multi-character Constant** コンポーネントを使用して赤、緑、青（RGB）の色値を出力します。

個々の RGB 値を設定するときは、下の 10 文字スキームを使用します。



以下の 2 桁の 16 進値で赤、緑、青の値を設定します。

- 16 進値の FF が 10 進値の 255 に等しい。
- 16 進値の 00 が 10 進値の 0 に等しい。

Color examples

定数	赤の値		緑の値		青の値		色
	16 進	10 進	16 進	10 進	16 進	10 進	
0xFF000000 COLOR	FF	255	00	00	00	00	Red（赤）
0x00FF0000 COLOR	00	00	FF	255	00	00	Green（緑）
0x0000FF00 COLOR	00	00	00	00	FF	255	Blue（青）
0xFFFF0000 COLOR	FF	255	FF	255	00	00	Yellow（黄）

Define Screen ページ / Priority（優先順位）プロパティについて

Inspector タブには複数のライブラリ項目が同時に **Signal** が True で **Enable** になったときに、どの項目を表示するかを決定する **Priority** プロパティがあります。

複数の項目が有効なとき、ディスプレイは **Priority** が高いほうの項目だけを表示します。

- **Priority** が **0** の項目は、**Priority** が **1** の項目の手前に表示されます。
- **Priority** が **1** の項目は、**Priority** が **2** の項目の手前に表示されます。

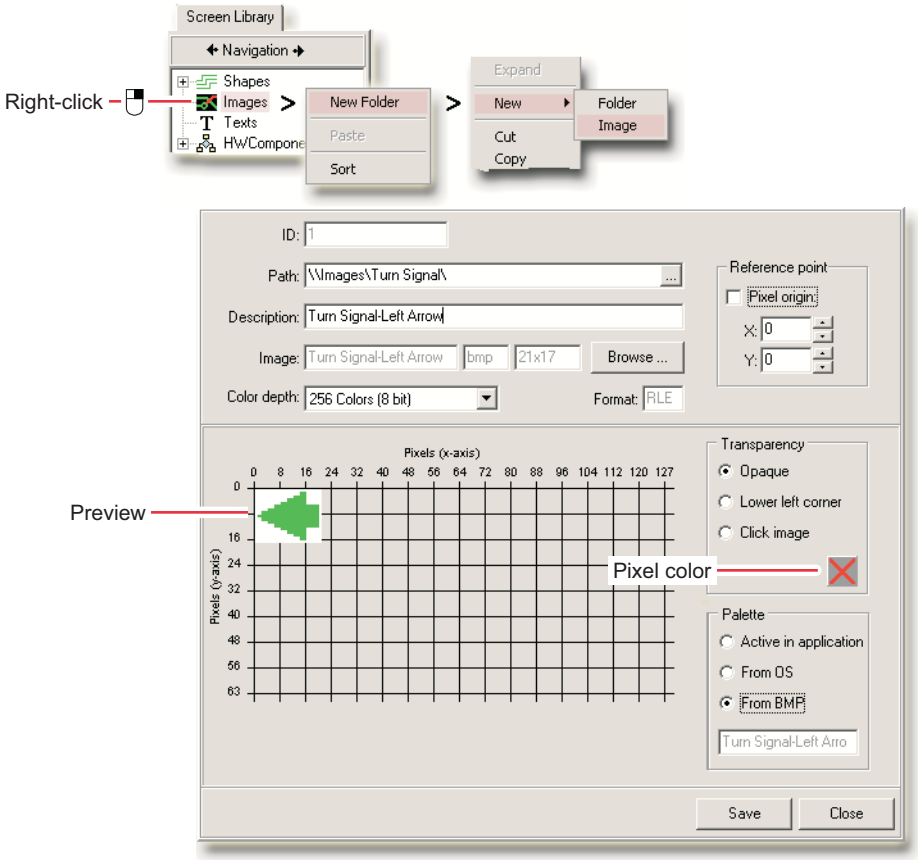
以下の例では、

- **Low Oil Pressure**、**Dirty Air Filter**、**Low Washer Fluid** の警告は、すべて同じスクリーン･エリアに表示されます。
- 警告の重要性は以下の順序です。
  - **Low Oil Pressure**（油圧が低い）
  - **Dirty Air Filter**（エアフィルタの汚れ）
  - **Low Washer Fluid**（洗浄液が少ない）
- このスクリーン･エリアに表示される警告は 3 つだけです。
- 警告はどれもビットマップで、赤色の背景に黄色のテキストです。

Priority の例

Alert	Enable Signal Status	Priority	Display		
Low Oil Pressure	True	0			
Dirty Air Filter	True	1		Low Oil Pressure	
Low Washer Fluid	True	2			
Low Oil Pressure	False	0			
Dirty Air Filter	True	1		Dirty Air Filter	
Low Washer Fluid	True	2			
Low Oil Pressure	False	0			
Dirty Air Filter	False	1		Low Washer Fluid	
Low Washer Fluid	True	2			

Define Screen ページ / Image Register（イメージの登録）



画像登録ペインを使用して、BMP イメージをグラフィック画面プロジェクトに使えるようにします。

画像の登録

- JPEG、TIFF、GIF イメージを BMP イメージファイルに変換します（元の JPEG、TIFF、GIF 画像は変更されません）。
- プロジェクトフォルダ外にある画像ファイルをプロジェクトフォルダにコピーします。
- **Color depth** などのプロパティを設定して、グラフィック画面へ画像がどのように表示されるかを決定します。

  Screen Library タブの **Images** ツリーから画像がプロジェクト内で使えるようにします。

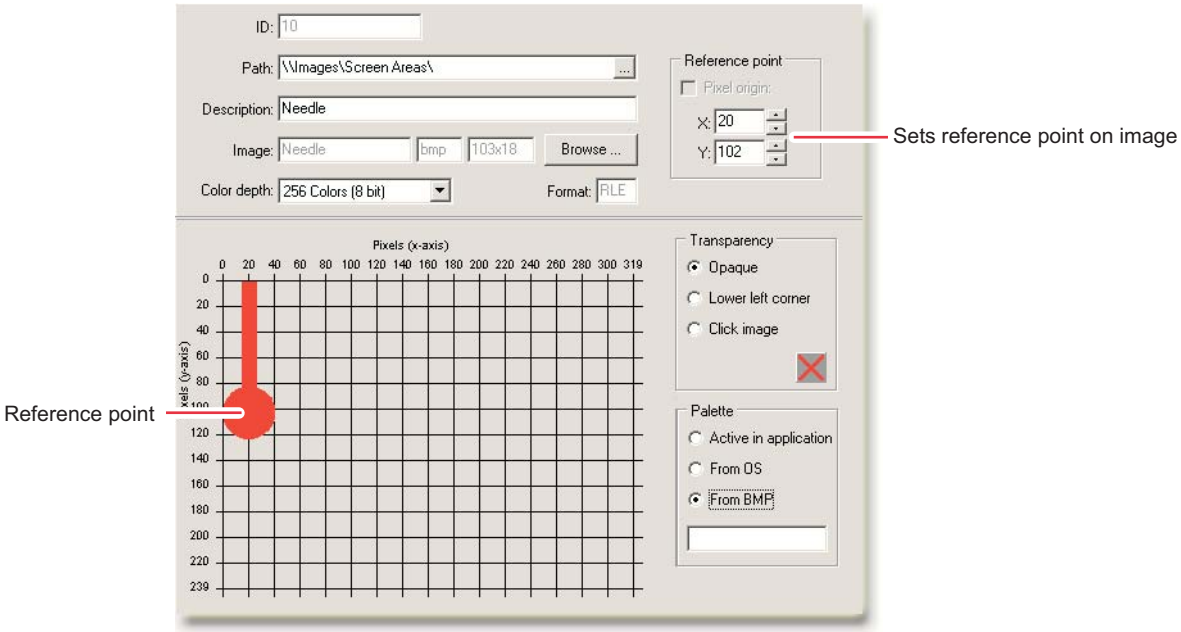
Define Screen ページ／Image Register

項目	説明
ID	識別に使用します。この値は変更できません。
Path（パス）	**Screen Library** タブの **Images** ツリーにある登録画像のロケーションを示します。
Description（説明）	登録する BMP 画像の参照名を入力します。 **Save** をクリックすると、**Images** ツリーにこの名前が表示されます。
Image（画像）	ソースファイル名、画像の種類、画素数を表示します。

Define Screen ページ／Image Register

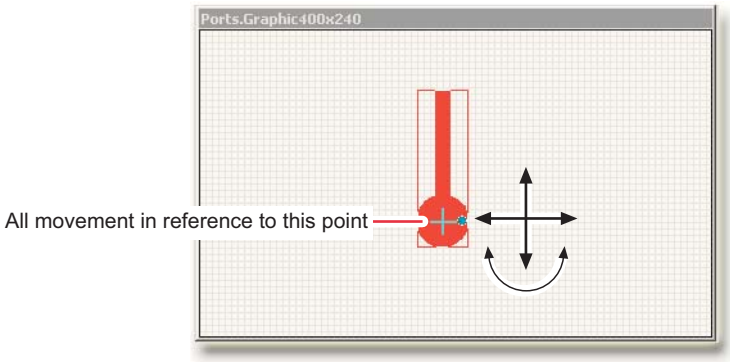
項目	説明
Browse（ブラウズ）	クリックすると、**Open** ウィンドウを表示します。 このウィンドウでソース BMP ファイルを見つけて選択します。スクリーンエディタが選択したファイルをグラフィックプロジェクトフォルダにコピーします。
Reference point （基準点）	**X** 軸、**Y** 軸の基準点を BMP 画像上に設定します。 BMP 画像の縦移動、横移動、回転はすべてこの点が基準になります。
Pixel origin （画素原点）	画像にユニークな色の画素がひとつあるときに使用できます。この画素を **Reference point** とするときにチェックマークを入れてください。
Color depth （色の深さ）	ビットマップ画像が表示する色の深さを設定します。
Format（フォーマット）	BMP 画像に用いる圧縮方法を表示します。
Preview（プレビュー）	BMP 画像のプレビューを表示します。
Transparency （透明性）	BMP 画像のどのエリアを透明にするかを設定します。 **Opaque**：BMP 画像の全画素が見えます。 **Lower left corner**： BMP 画像の左下隅の画素と同色のすべての画素が透明になります。 **Click image**： クリックする画素と同色のすべての画素が透明になります。 **Pixel color**：選択した画素の色を表示します。
Palette（パレット）	BMP 画像が使用するカラーパレットを設定します。 **Active in application**：アプリケーションで以前に定義したカラーパレットを使用します。 **From OS**：ハードウェア OS で定義したカラーパレットを使用します。 **From BMP**： BMP ソースファイルのカラーパレットを使用します。
Save（保存する）	BMP 画像を登録します。 BMP ソースファイルをプロジェクトフォルダにコピーします。**Screen Library** タブの **Images** ツリーに登録画像名を表示します。
Close (閉じる)	画像登録ペインを閉じます。

Define Screen ページ / Reference Point（基準点）について



Reference point は画像上に **X** 軸と **Y** 軸による基準点を設定します。

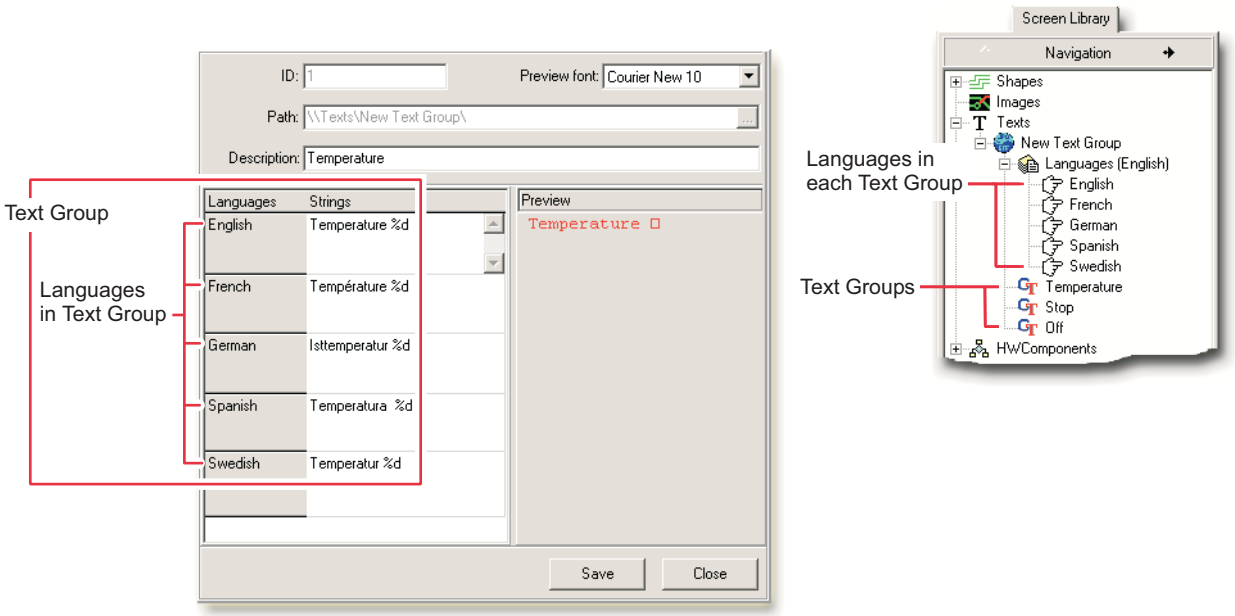
デフォルトの **Reference point** は画像の左上隅です。



レイアウトペインの画像をクリックすると、小さな青十字が **Reference point** を示します。

画像の縦移動、横移動、回転はすべてこの点が基準になります。

Define Screen ページ / Text Register（文字列の登録）



Text Register（文字列登録）ペインを使用して、**Screen Library** タブの **Texts** ツリーに **Text Group** を追加します。

Text Group の登録

- **Text Group** の文字列をグラフィックプロジェクトフォルダのファイルにコピーします。
- 文字列の表示特性を定義（登録）します。
- **Screen Library** タブの **Texts** ツリーに **Text Group** の **Description** 名を表示します。

Define Screen ページ / 文字列登録ペイン

項目	説明
ID	識別に使用します。この値は変更できません。
Preview font（プレビューフォント）	**Preview** ペインに表示する文字列のフォントとサイズを設定します。 （これはディスプレイに表示するフォントを設定するものではありません。ディスプレイに表示するフォントは、**Inspector** タブの **Font** プロパティで設定します。）
Path（パス）	**Screen Library** タブの **Texts** ツリーにある **Text Group** のロケーションを示します。
Description（説明）	**Text Group** の名前を入力します。**Save** をクリックすると、この名前が **Screen Library** タブの **Texts** ツリーに表示されます。 テキスト登録ペインを開いてグループの文字列を表示したいときは、**Texts** ツリーの **Text Group** 名をクリックします。
Languages（言語）	**Text Group** の文字列を表示できる諸言語を一覧表示します。
Strings（文字列）	ここに表示したいテキストを入力します。 改行には Ctrl+Enter を使用します。 文字列中のデータ値を表示するときは、%印刷文字を使用します。

Define Screen ページ / 文字列登録ペイン

項目	説明
Preview pane（プレビューペイン）	入力した **String** をプレビューできます。 （ディスプレイに表示する文字列のフォントは、**Inspector** タブの **Font** プロパティで設定します。）
Save （保存する）	**Text Group** を登録します。 グループも文字列をグラフィックプロジェクトフォルダのテキストファイルにコピーします。 **Screen Library** タブの **Texts** ツリーに、**Text Group** の **Description** を表示します。
Close (閉じる)	文字列登録ペインを閉じます。

アプリケーションデータログ機能

この章では **Define APPAreas** コンポーネントと **Define APPLog** コンポーネントについて記述します。これらの二つのブロックにより基本的なアプリケーションデータログ機能を実装します。

アプリケーションデータログ機能はハードウェアに依存します。全ての PLUS+1 コントローラがアプリケーションデータログ機能をサポートしているわけではありません。

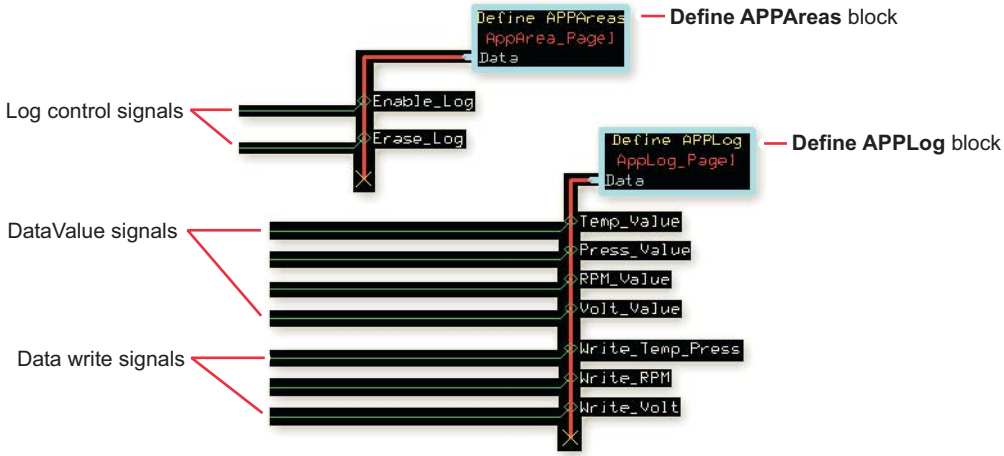
アプリケーションデータログではザウアーダンフォス社 PLUS+1 コントローラのメモリーにデータを書き込みます。PLUS+1 サービスツールプログラムでこのデータにアクセスします。サービスツールプログラムはまず暗号化された P1A（PLUS+1 アプリケーションデータログ）ファイルにデータを書き込み、アクセス権に従って CSV（カンマ区切り）形式のファイルに出力します。

ユーザは **Access** コンポーネントを用いてサービスツールプログラムのアプリケーションデータログとその内容へのアクセスを制限することができます。これらのコンポーネントの使用方法については ***320*** ページの ***Access*** を参照してください。

この章の内容

アプリケーションデータログ機能 396
アプリケーションデータログ機能の基本的な要素 397
Define APPAreas Block 398
Define APPAreas ブロック/Inspector タブ 400
Define APPLog ブロック 403
Define APPLog Block/Add Texts 404
Define APPLog ブロック / Text Register Pane（文字列登録ペイン） 405
Define APPLog ブロック / Inspector タブ — データ書き込みプロパティ 406
Define APPLog ブロック / Inspector タブ — DataValue プロパティ 407
アプリケーションデータログの内容を読み取る方法 408

アプリケーションデータログ機能の基本的な要素



これはどのように **Define APPAreas** ブロックと **Define APPLog** ブロックを使用して基本的なアプリケーションデータログ機能を実装するかを示しています。

- **Define APPAreas** ブロックは、
  - アプリケーションデータログの有効化や、アプリケーションデータログの内容の消去をコントロールする Log control signal を受信します
  - データアプリケーションログのためのコントローラのメモリーを予約し、ログタイプをサーキュラかリニアかのどちらかに設定します。
- **Define APPLog** ブロックは、
  - アプリケーションデータログに書き込む値を含む DataValue 信号を受信します。
  - アプリケーションデータログをいつ書き込むかを決定する Data 書き込み信号を受信します。
  - Has properties that tag data by category and access level.

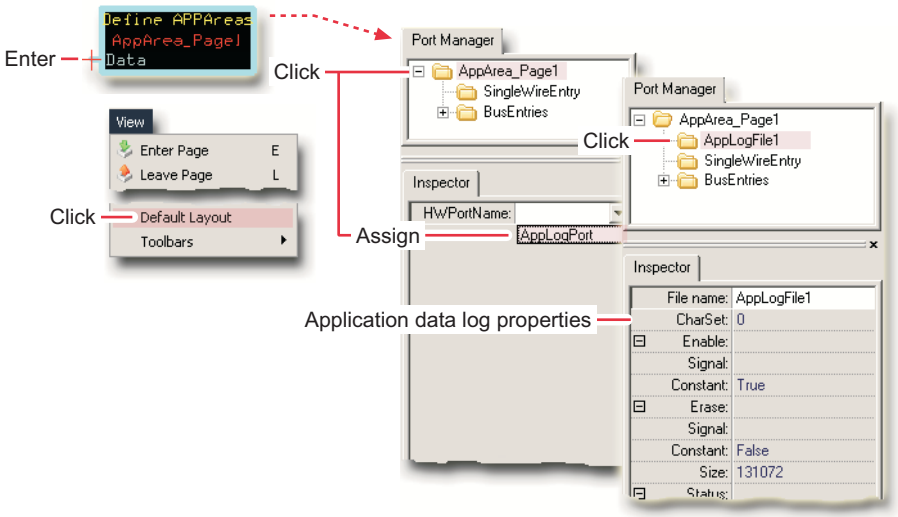
アプリケーションは複数のアプリケーションデータログを持つことができます。

この図には Log コントロール、DataValue、Data 書き込み信号については示していません。

Basic Elements of Application Data Logging

Item	Description
Define APPAreas ブロック	コントローラのメモリをアプリケーションデータログのために予約します。 アプリケーションデータログの型をサーキュラ型かリニア型のどちらかに定義します。 コントローラがいつデータをアプリケーションデータログに記録するかを定義します。 コントローラがいつアプリケーションデータログの内容を消去するのかを定義します。
Define APPLog ブロック	どのデータをアプリケーションデータログに書き込むのかを定義します。 いつデータをアプリケーションデータログに書き込むのかを定義します。

Define APPAreas Block



この図は **Define APPAreas** ブロックに入り、**Default Layout** ビューを選択し、アプリケーションデータログに **HWPortName** をアサインしたときに表示される要素を示しています。

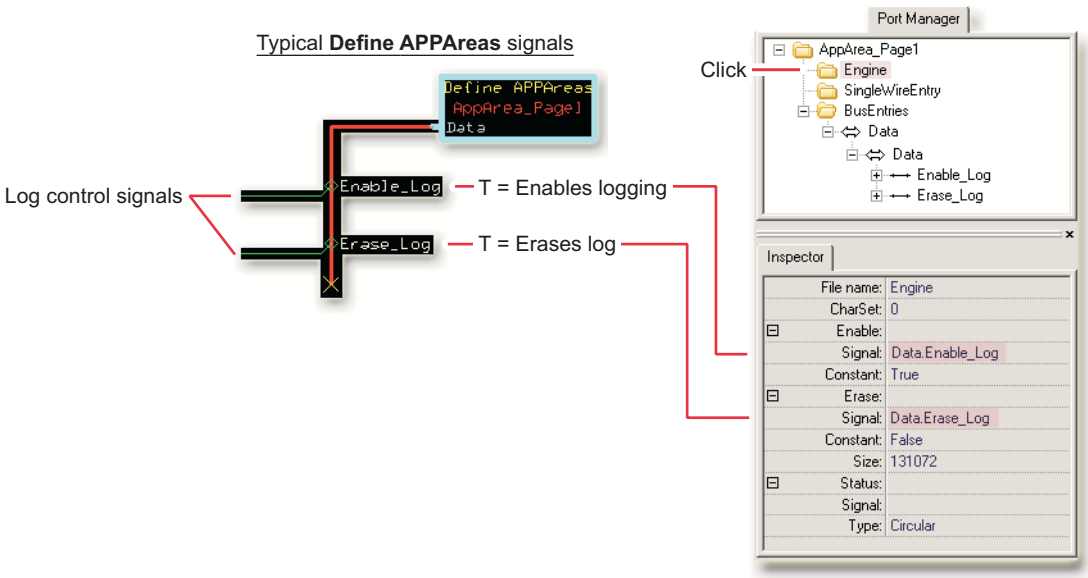
- **Define APPAreas** ブロックは、
  - アプリケーションデータログ用にコントローラのメモリを予約します。
  - コントローラがいつ以下のことを行うかを定義します
    - データをアプリケーションデータログに記録できる
    - アプリケーションデータログの内容を消去する
  - アプリケーションデータログの型をサーキュラ型かリニア型のどちらかに定義します。
- Screen Library タブも表示されますが、Define APPAreas ブロック内では使用しません。
- **Define APPAreas** ブロックはサブバスをサポートしていません。信号を **Define APPAreas** ブロックに送るにはメインバスを使用してください。

アプリケーションデータログを定義する前に **Inspector** タブ内の **HWPortName** をアサインしておく必要があります。

Define APP Areas ブロック

項目	説明
Port Manager タブ	このタブを使用してアプリケーションデータログのプロパティを管理します。 **AppArea_Page1** フォルダ：クリックするとアプリケーションデータログで使用する **HWPortName** を表示します。 **AppLogFile1** フォルダ：アプリケーションデータログのデフォルトのファイル名です。PLUS+1 サービスツールからこの名前でアプリケーションデータログにアクセスします。**Inspector** タブを使用してこの名前を変更します。 **SingleWireEntry** フォルダ：ここでは使用しません。 **BusEntries** フォルダ：クリックすると **Define APPAreas** ブロックの **Data** 入力につながる信号のツリービューを表示します。
Inspector タブ	このタブを使用してアプリケーションデータログのプロパティを見たり管理したりします。
Buttons	PLUS+1 GUIDE ウィンドウのサイズにより、どのボタンが表示されるかが決まります。
	スクリーンエディタの終了：クリックして **Define APPAreas** ブロックを終了します。
	ヘルプ・コンテンツ：PLUS+1 GUIDE のヘルプを開きます。

Define APPAreas ブロック/Inspector タブ



Inspector タブを使用してアプリケーションデータログのプロパティを管理します。

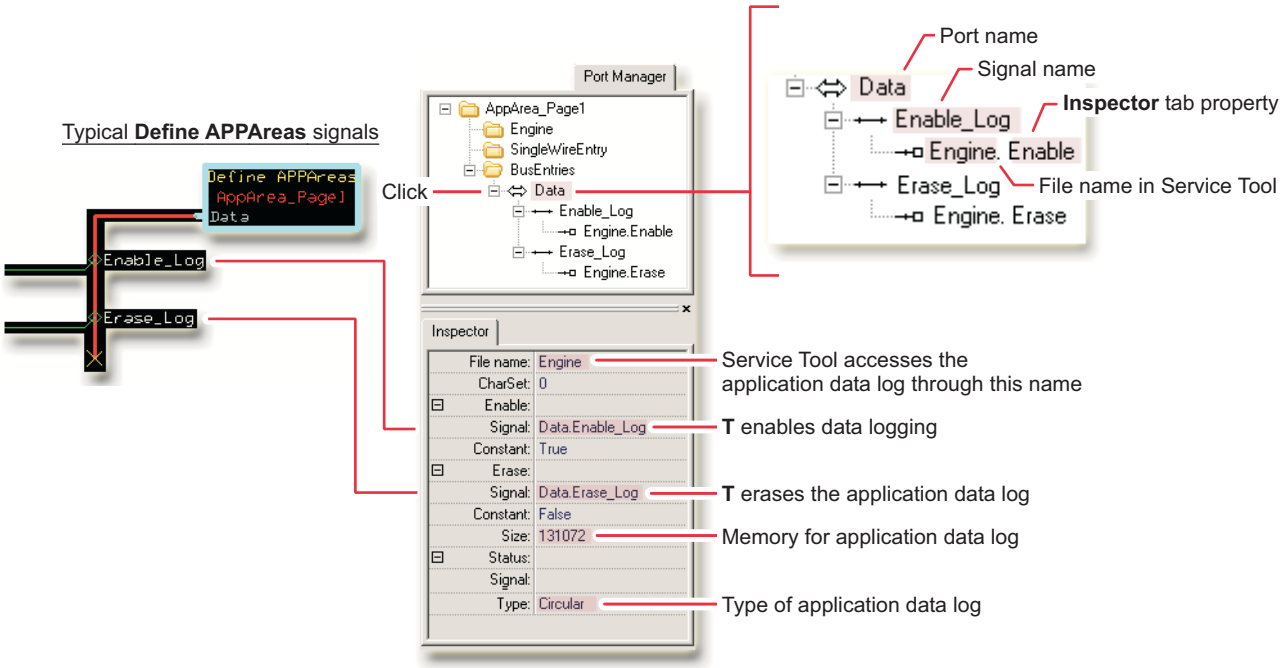
Define APPAreas ブロック/Inspector タブ

項目	説明
File name	PLUS+1 サービスツールがこの名前を用いてアプリケーションデータログにアクセスします。分かりやすい名前を付けてください。
CharSet	アプリケーションデータログに書き込むデータのキャラクタセットを定義します。 この値を **0** に設定すると Western, Roman アルファベットを用います。 非ローマンアルファベットを使用するデータには必要なキャラクタセット番号を入力してください。
Enable	T の場合、データをアプリケーションデータログに書き込むことができます。 アプリケーションデータログにデータを書き込む前にこのプロパティを T にしておく必要があります。 **Signal** 入力は **Constant** 値より優先します。 T = データをアプリケーションデータログに書き込むことができます。 F = データをアプリケーションデータログに書き込むことができません。
Erase	T の場合、アプリケーションデータログの内容を消去します。 **Signal** 入力は **Constant** 値より優先します。 T = アプリケーションデータログの内容を消去します。
Status	アプリケーションデータログのステータスを出力する U16 信号を選択します。 ビット 1 (0x0001) = アプリケーションデータログにデータを書き込んでいます。 ビット 2 (0x0002) = アプリケーションデータログの内容をアプリケーションデータログファイルに書き込んでいます。読み込みプロセスが終了すると保留中のデータがアプリケーションデータログに書き込まれます。 ビット 3 (0x0004) = アプリケーションデータログの内容を消去しています。消去プロセスが終了すると保留中のデータがアプリケーションデータログに書き込まれます。 ビット 4 (0x0008) = アプリケーションデータログがいっぱいです。アプリケーションデータログがリニア型の場合のみ適用されます。

Define APPAreas ブロックInspector タブ

項目	説明
Size	アプリケーションデータログのメモリサイズを設定します。アプリケーションデータログへのメモリ割り当ての詳細についてはコントローラの API 仕様書を参照してください。
Type	アプリケーションデータログを **Circular**（循環）型か **Linear**（リニア）型のどちらかに定義します。

Define APPAreas ブロック / Port Manager とInspector タブについて



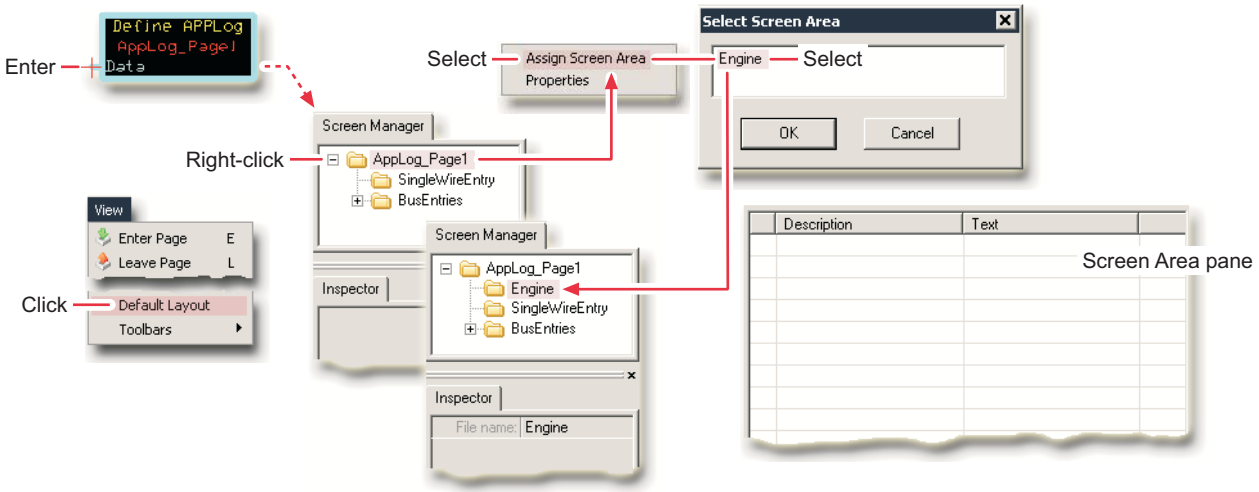
この図では以下のことを示しています。

- **Define APPAreas** ブロックの **Data** 入力へ **Enable_Log** および **Erase_Log** のブール信号を配線します。
- **Define APPAreas** ブロックの **Port Manager** と **Inspector** タブにおいて、
  - **Enable** プロパティをコントロールするために **Enable_Log** 信号を選択します。

    アプリケーションデータログにデータを書き込む前に **Enable_Log** 信号を **T** にしておく必要があります。
  - **Erase** プロパティをコントロールするために **Erase_Log** 信号を選択します。

    Erase_Log 信号が **T** のとき、アプリケーションデータログを消去します。
- ブロックへの **BusEntries**（信号）がどのように **Inspector** タブのプロパティにマッピングされているかを示しています。

Define APPLog ブロック



この図は、**Define APPLog** ブロックに入り、**Default Layout** ビューを選択し、スクリーンエリアを選択したアプリケーションデータログに割り当てたときに表示されるエレメントを示しています。

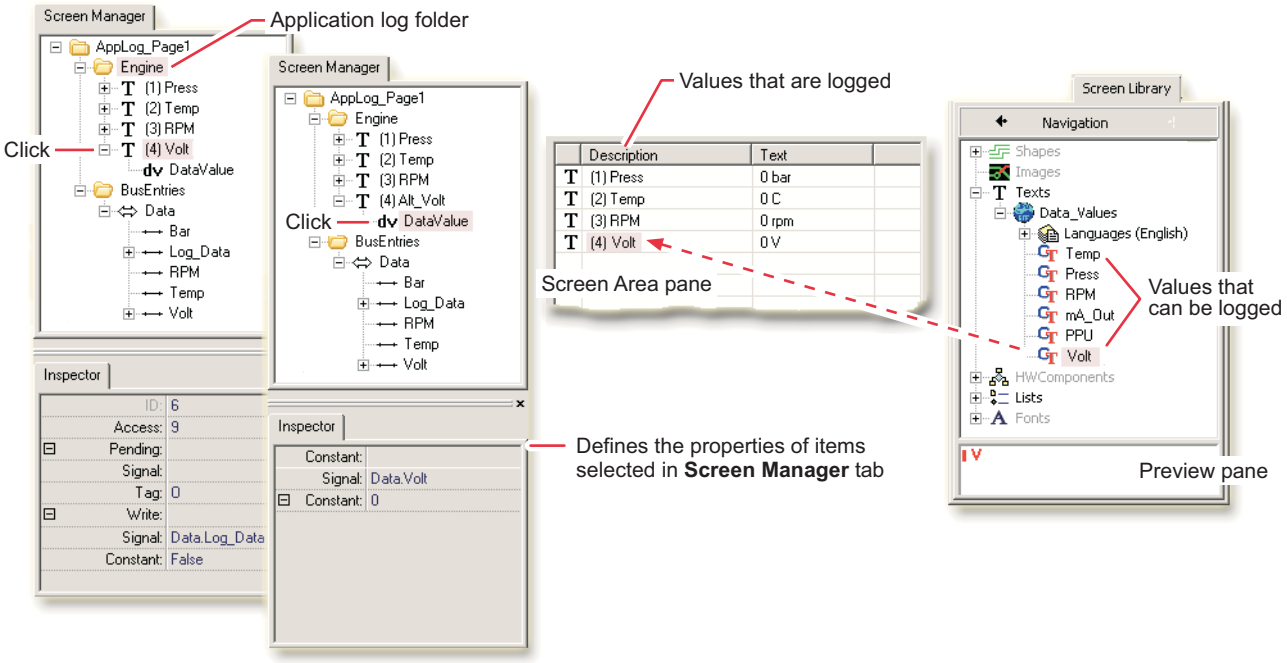
- **Define APPLog** ブロックはサブバスをサポートしていません。このブロックに信号を入れるにはメインバスを使用してください。

File メニューの **Import Page** コマンドと **Import Block** コマンドはこのコンポーネントでは使用できません。**Define APPLog** ブロックをインポートするとその内容が欠落します。

Define APPLog ブロック

項目	説明
Screen Manager タブ	アプリケーションデータログのフォルダ（この図では **Engine** フォルダ）を表示します。
スクリーンエリアペイン	アプリケーションデータログに書き込まれる **Texts**（値）をリスト表示します。Screen Area に選択したアプリケーションデータログを割り当てたら空のペインが表示されます。
Inspector タブ	アプリケーションデータログに書き込む値のプロパティを管理します。
Buttons	PLUS+1 GUIDE ウィンドウのサイズにより、どのボタンが表示されるかが決まります。
	スクリーンエディタの終了：クリックして **Define APPLog** ブロックを終了します。
	ヘルプ・コンテンツ：PLUS+1 GUIDE のヘルプを開きます。

Define APPLog Block/Add Texts



アプリケーションデータログフォルダ（ここでは **Engine** フォルダ）の内容によって以下のことが決まります。

- どのデータをアプリケーションデータログに書き込むか
- いつデータをアプリケーションデータログに書き込むか

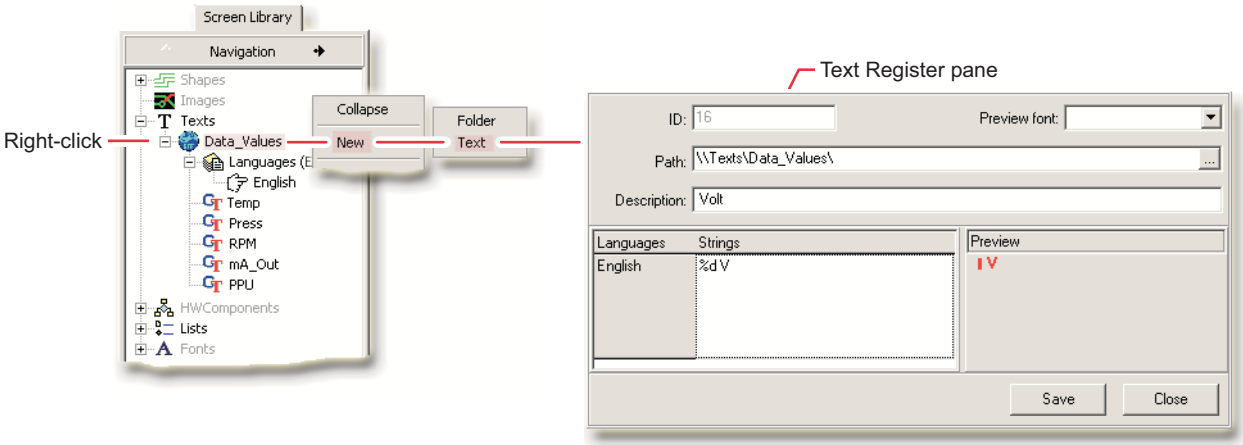
アプリケーションデータログフォルダの内容は以下のように定義します。

- 記録する値の **Texts** を **Screen Library** タブからスクリーンエリアペインにドラッグします。
- **Inspector** タブを使用して **Texts** のプロパティを定義します。

Define APPLog ブロック / Texts の追加

項目	内容
Screen Manager タブ	**AppLog_Page1** フォルダ：クリックして **APPLog_Page1** プロパティのツリービューを表示します アプリケーションログフォルダ（この例では **Engine**）：クリックしてスクリーンエリアに追加された **Texts** のツリービューを表示します。 **SingleWireEntry** フォルダ：ここでは使用しません。 **BusEntries** フォルダ：クリックして **Define APPLog** ブロックの **Data** 入力に配線されている全ての信号のツリービューを表示します。
Inspector タブ	**Screen Manager** タブで選択したアイテムのプロパティを定義します。
スクリーンエリアペイン	アプリケーションデータログに書き込む値の **Texts** をリスト表示します。このペインの **Description** 列に表示されている **Texts** の順序がアプリケーションデータログに現れる値の順序を決定します。
Screen Library タブ	アプリケーションデータログに書き込むことができる **Texts** の値のツリービューを表示します。 アプリケーションデータログに書き込まれる値を選択するには、**Screen Library** タブから **Texts** をスクリーンエリアペインにドラッグします。
プレビューペイン	**Texts** 値がデータログファイル内でどのように現れるかをプレビュー表示します。

Define APPLog ブロック / Text Register Pane（文字列登録ペイン）



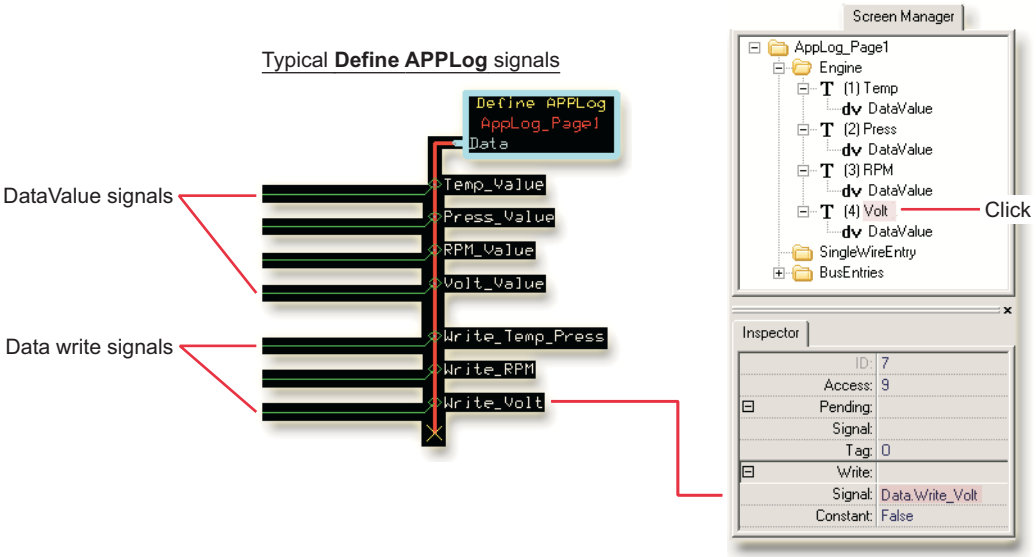
Text Register pane（文字列登録ペイン）を使用して **Screen Library** タブに **Texts** を作成します。

Texts を作成してアプリケーションデータログに書き込まれる文字列を定義します。C

Define APPLOG ブロック / Text Register Pane（文字列登録ペイン）

項目	内容
ID	ID のために使用されます。ユーザは変更できません。
Preview font	使用しません。
Path	**Screen Library** タブの **Texts** ツリー内でのパスです
Description	**Texts** アイテムの記述的名称です。 ここで入力した名前は **Screen Library** タブの **Texts** ツリーおよび、このアイテムをスクリーンエリアペインにドラッグしたときには **Screen Manager** タブのアプリケーションデータログフォルダに表示されます。
Languages	**Texts** アイテムの言語を区別します。
Strings	アプリケーションデータログに書き込まれる文字列を入力します。 %印刷文字を用いてデータ値を文字列中に表示します。
プレビューペイン	**Strings** エントリーのアプリケーションデータログ内でのプレビューを表示します。
Save	文字列登録ペインを閉じて、**Screen Library** タブの **Texts** ツリーと **Screen Manager** タブのアプリケーションデータログフォルダに **Texts** を追加します。
Close	文字列登録ペインを閉じます。

Define APPLog ブロック / Inspector タブ — データ書き込みプロパティ



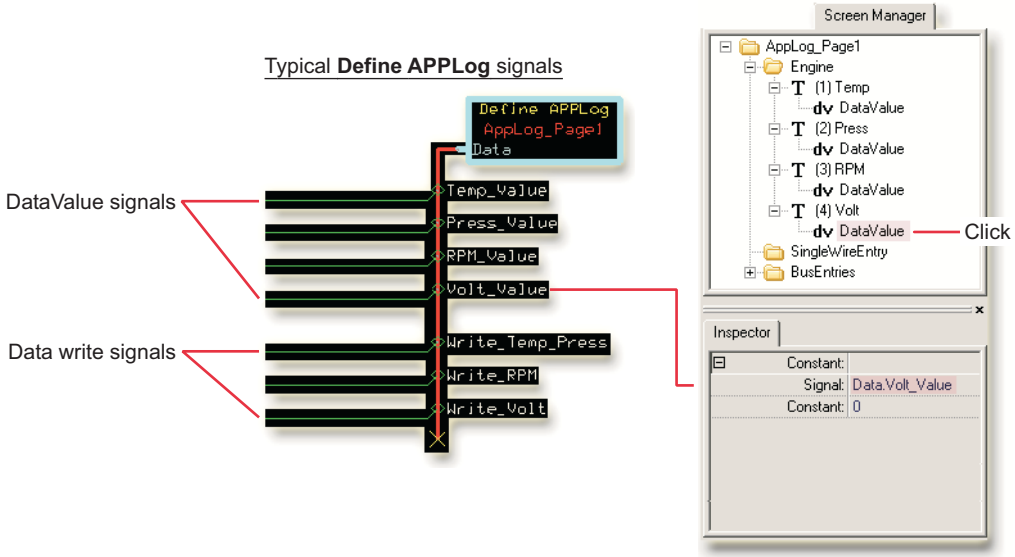
データの書き込み（Write）プロパティにより、以下のことを定義します。

- いつデータをアプリケーションデータログに書き込むか
- データのカテゴリとアクセスレベル

Define APPLog ブロック / Inspector タブ　データ書き込みプロパティ

項目	説明
ID	内部的な識別のために用いられる読み取り専用の値
Access	データのアクセスレベルを設定します。 この値は **Access** コンポーネントと組み合わせて用い、PLUS+1 サービスツールを使用したアプリケーションデータログへのユーザのアクセスを制限します。
Pending	データの書き込み準備の状態を示すブール値を選択します。 T = データはアプリケーションデータログへの書き込み準備が完了している
Tag	データのカテゴリを設定します。 この値は **Access** コンポーネントと組み合わせて用い、PLUS+1 サービスツールを使用したアプリケーションデータログ内のデータのカテゴリへのユーザのアクセス権を制限します。 **E** = エラー（Error） **S** = 統計値（Statistics） **O** = その他（Other）
Write	T の場合、データをアプリケーションデータログに書き込みます。 **Signal** 入力は **Constant** 値より優先します。 T = データをアプリケーションデータログへ書き込み.

Define APPLog ブロック / Inspector タブ — DataValue プロパティ



DataValue プロパティはアプリケーションデータログに書き込まれる値を定義します。

Define APPLog ブロック / Inspector タブ　DataValue プロパティ

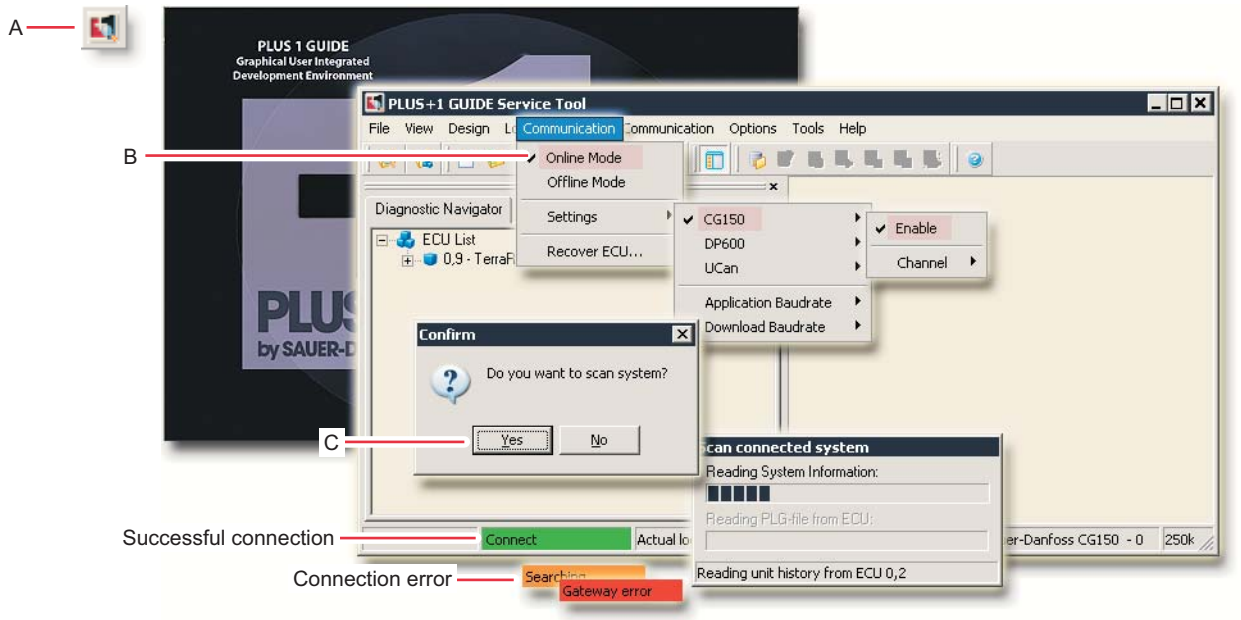
項目	内容
Constant	アプリケーションデータログに書き込む値を持つ信号を選択します。

アプリケーションデータログの内容を読み取る方法

PLUS+1 サービスツールプログラムを使用してアプリケーションデータログの内容にアクセスします。

以後の手順は **Access** プロパティおよび **Access** コンポーネントがアプリケーションデータログの内容へのアクセス制限をしていないアプリケーションからデータをアクセスする手順を示しています。

1. PLUS+1 コントローラと PC の通信に必要なハードウェアおよびケーブルをセットアップします。



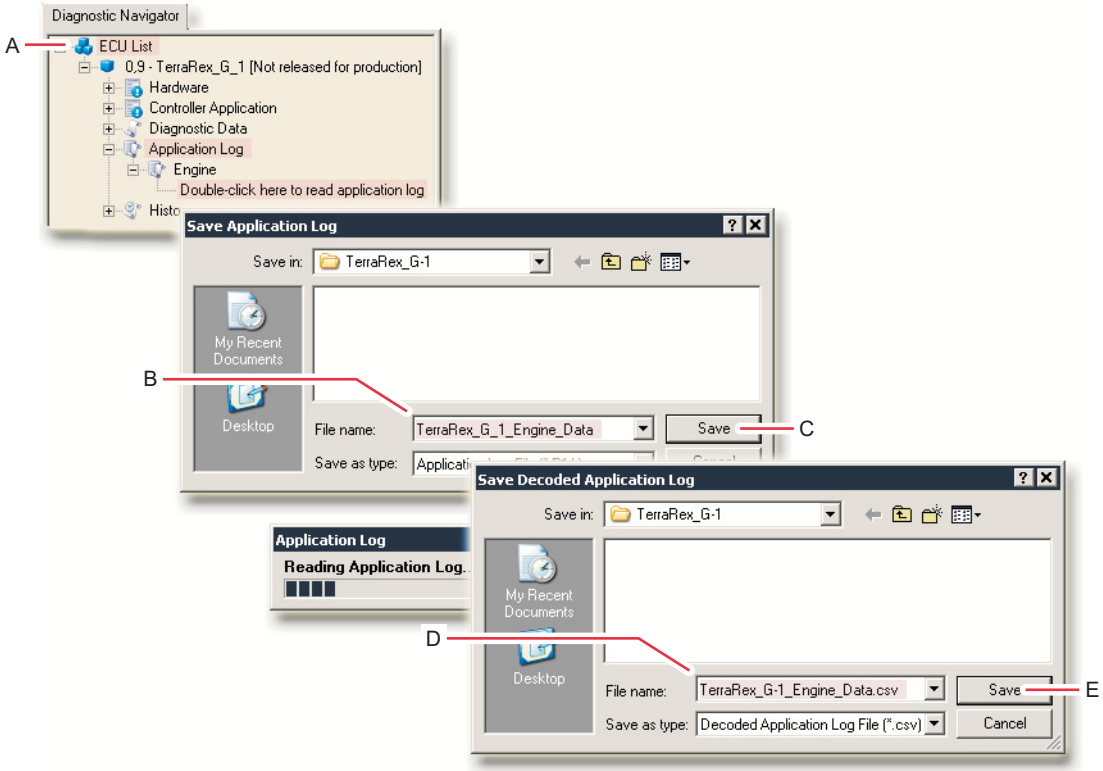
2. PLUS+1 サービスツールプログラムを起動します。

   A. **PLUS+1 GUIDE** ウィンドウツールバーで、PLUS+1 Service Tool ボタンを押します。

   PC の **Start** メニューからこのプログラムを起動することもできます。

   B. サービスツールプログラムの **Communications > Settings** メニューで **CG150** が選択されていることを確認します。

   C. **Confirm** ウィンドウが表示されたら **Yes** をクリックしてハードウェアをスキャンします。



3. アプリケーションデータログの内容を CSV（カンマ区切り）形式のファイルに書き込みます。

   A. **Diagnostic Navigator** タブツリーで、以下のようにクリックします。

      ECU List > **Application Log** > **Double click here to read application log**

      Save Application Log ウィンドウが開きます。

   B. **Save Application Log** ウィンドウで、アプリケーションログファイル(*.P1A)の名前を入力します。

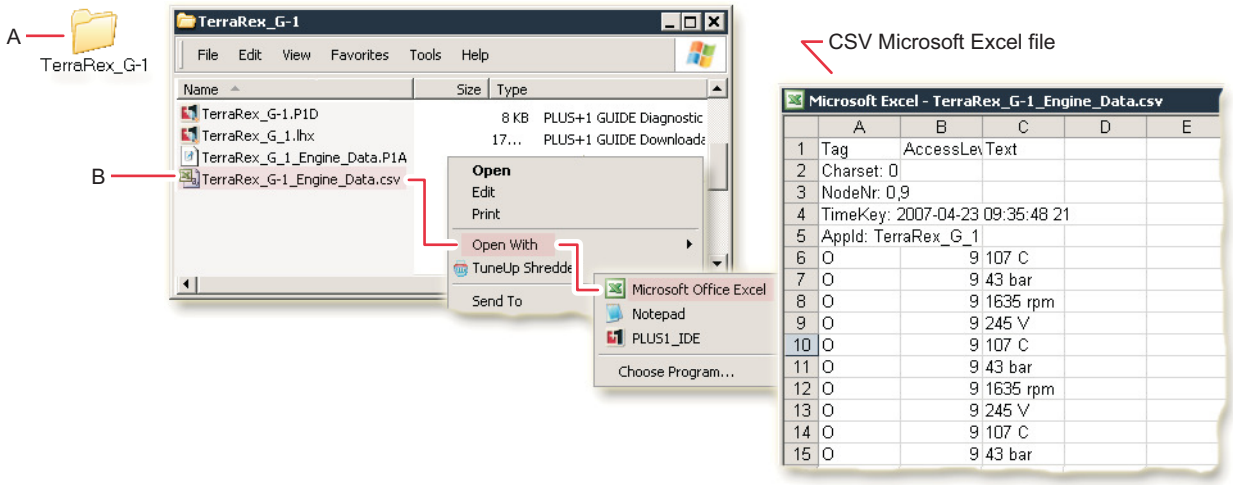
   C. **Save Application Log** ウィンドウで、**Save** をクリックしてアプリケーションデータログの内容をアプリケーションログファイルに書き込みます。

      （アプリケーションログファイル(*.P1A)は暗号化されたファイルです。このファイルを解読するユーザ権限は **Define APPLog** ブロックの **Access** プロパティを **Access** コンポーネントと共に使用することで制限することが可能です。）

      Save Decoded Application Log ウィンドウが表示されます。

   D. **Save Decoded Application Log** ウィンドウで、CSV ファイルの **File name** を入力します。

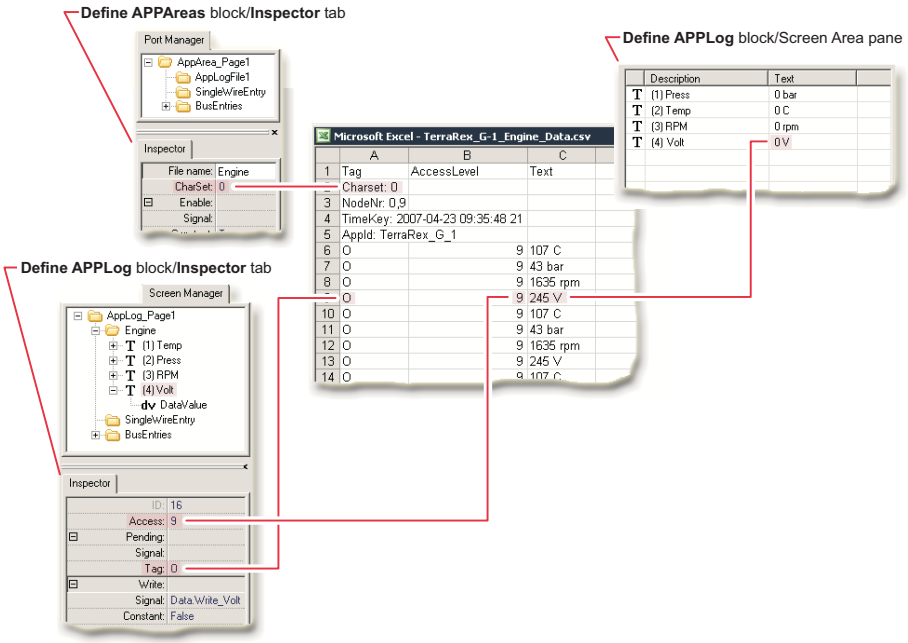
   E. **Save** をクリックします。



4. CSV アプリケーションログファイルを開きます。

   A. CSV アプリケーションログファイルを含むフォルダをブラウズします。

   B. CSV アプリケーションログファイルを Microsoft Notepad（メモ帳）または Microsoft Excel で開きます。



この図は **Define APPAreas** ブロックや **Define APPLog** ブロックのプロパティが CSV アプリケーションデータログ内のデータの表示を決定している様子を示しています。

PLUS+1 の手引き

本章では、PLUS+1 GUIDE アプリケーションの基本的な操作手順を説明します。

この章の内容

PLUS+1 の手引き 411
選択 412
間違いの取消し 412
マウスクリックによるズーム操作.....413
マウスホイールによるズーム操作 413
キーボードによる拡大・縮小 413
単一アイテムの削除.....414
複数アイテムの削除.....415
ページの全内容を他のページにコピー 416
信号－バス接続の削除 418
表示の更新.....419
ページ全体の表示 419
右クリック＆ドラッグによる表示のパン.....420
右クリックによる表示のパン 420
ボタンによるアプリケーションのナビゲーション 421
Page Navigator タブによるアプリケーションのナビゲーション 421
ペインの表示と非表示.....422
ハードウェア記述のインストール 423
Inspector タブによるプロパティの変更.....425
ポップアップ編集ウィンドウによるプロパティの変更 426
ページの新規作成 427
View Logical Net の使用 438
Read-only Parameters（読み取り専用パラメータ）ファイルの作成.....443
Readme LHX ファイルの追加 464
パーツ番号やシリアル番号によるダウンロードの制限.....466
アプリケーション内の変数へのサービスツールによるアクセスの制限.....468
リンクページの作成 471
リンクページのオブジェクトページへの変換 477
リンクページとオブジェクトページで作業する 478
ページの Name Space の表示.....486
ページビューの無効化（ロック）.....489
コメントのフォントと色をカスタマイズする 496
プラグインコードの追加.....500

選択

Windows アプリケーションのパラダイムはオブジェクトーアクションで、まずオブジェクトを選択してからそのオブジェクトに適用するアクションを選択します。

PLUS+1 GUIDE アプリケーションのパラダイムはアクションーオブジェクトで、まずアクションを選択してからそのアクションのオブジェクトを選択します。

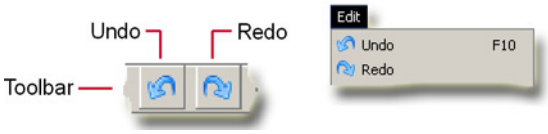
PLUS+1 GUIDE アプリケーションでは、

1. クリックしてアクションを選択します。
2. アクションを実行します。
3. 別のアクションをクリックするか Esc を押して、アクションを停止します。

例: 図面エリアに **3 Digit Autotype**（3 桁オートタイプ）定数コンポーネントを追加するとき:

1. **Component** タブの **Constant** 欄で、**3 Digit Autotype** をクリックする。
2. 図面エリアでクリックするたびに、**3 Digit Autotype** コンポーネントが追加されます。
3. 別のアクションをクリックするか Esc を押して、コンポーネントの追加を止めます。

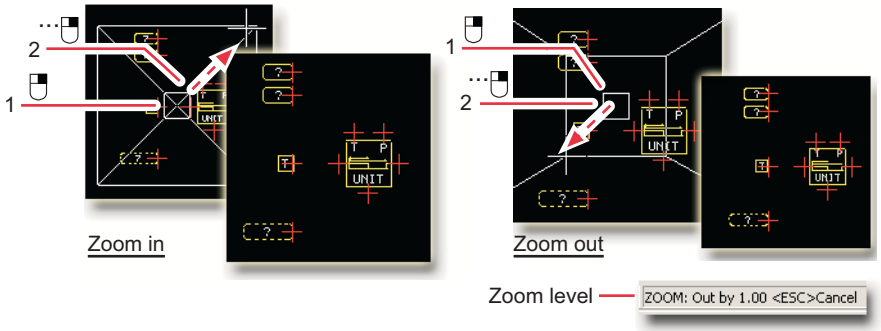
間違いの取消し



間違いを訂正するときは、ツールバーの Undo（取消し）ボタンと Redo（やり直し）ボタンか、セットアップメニューの **Undo** コマンドと **Redo** コマンドを使用します。

デフォルトでは、PLUS+1 GUIDE アプリケーションは **Undo**、**Redo** 機能が有効になっています。**Options** ウィンドウで、これらの機能の有効、無効を設定します。
このウィンドウへのパスは、**Setup** メニュー> **Options** > **Options** ウィンドウ > **General** です。

マウスクリックによるズーム操作



マウスを用いて拡大・縮小するには：

1. ズームの中心を右クリックします。
2. 右クリック＆ドラッグで、拡大または縮小します。
   - 拡大するときは、右上に対角的にドラッグします。
   - 縮小するときは、左下に対角的にドラッグします。

   ステータスバーに拡大・縮小レベルが表示されます。

マウスホイールによるズーム操作

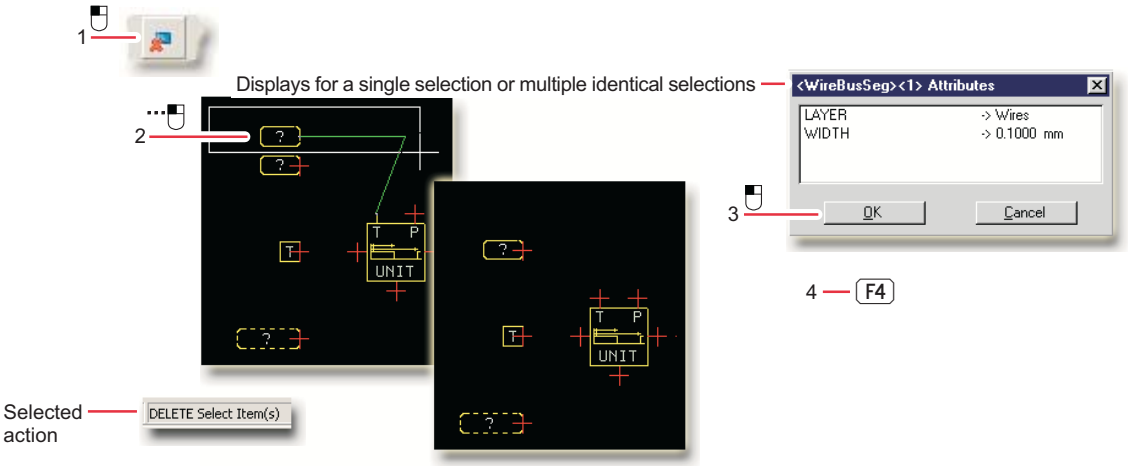
図面エリア上でマウスホイールを回転させてズームインとズームアウトを行います。

キーボードによる拡大・縮小

キーボードを用いて拡大・縮小するには：

1. ズームの中心を右クリックします。
2. 拡大または縮小します。
   - PgUp で拡大します。
   - PgDn で縮小します。

単一アイテムの削除



1. ツールバーの **Delete** ボタンをクリックします。

   ステータスバーに削除を選択したことが表示されます。

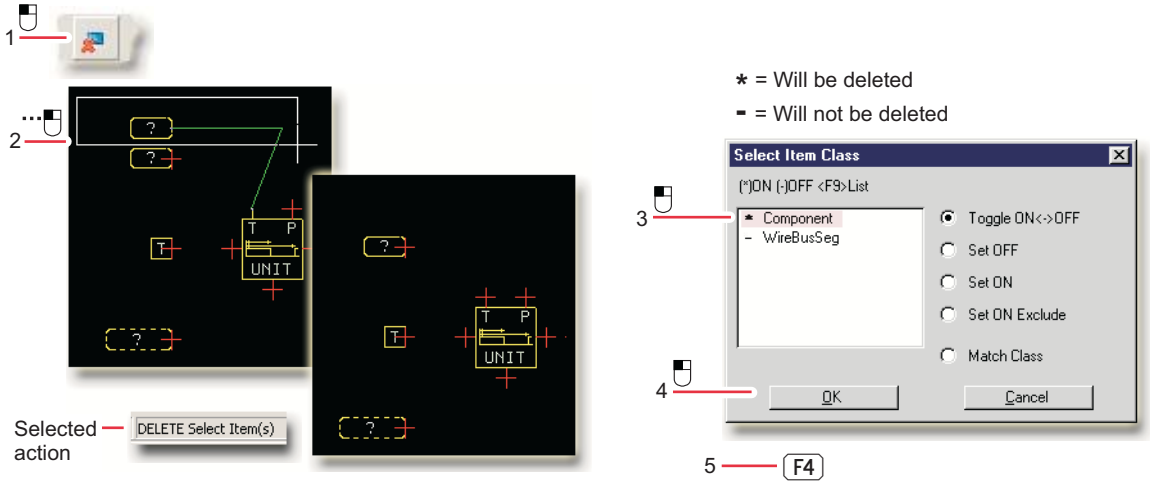
2. 図面エリアで、削除したい項目をクリック＆ドラッグで選択します。

   1 項目を選択するか、同一項目を複数選択（3 コンポーネントの選択など）すると、**Attributes** ウィンドウが開きます。

3. **OK** をクリックして選択した項目を削除します。

4. F4 を押して図面エリアの表示を更新します。

複数アイテムの削除



1. ツールバーの **Delete** ボタンをクリックします。

   ステータスバーに削除を選択したことが表示されます。

2. 図面エリアで、複数項目をクリック＆ドラッグで選択します。

   複数のクラスに属する複数項目（コンポーネントとワイヤなど）を選択すると、**Select Item Class** ウィンドウが開きます。

3. **Select Item Class** ウィンドウで削除したい項目を選択します。

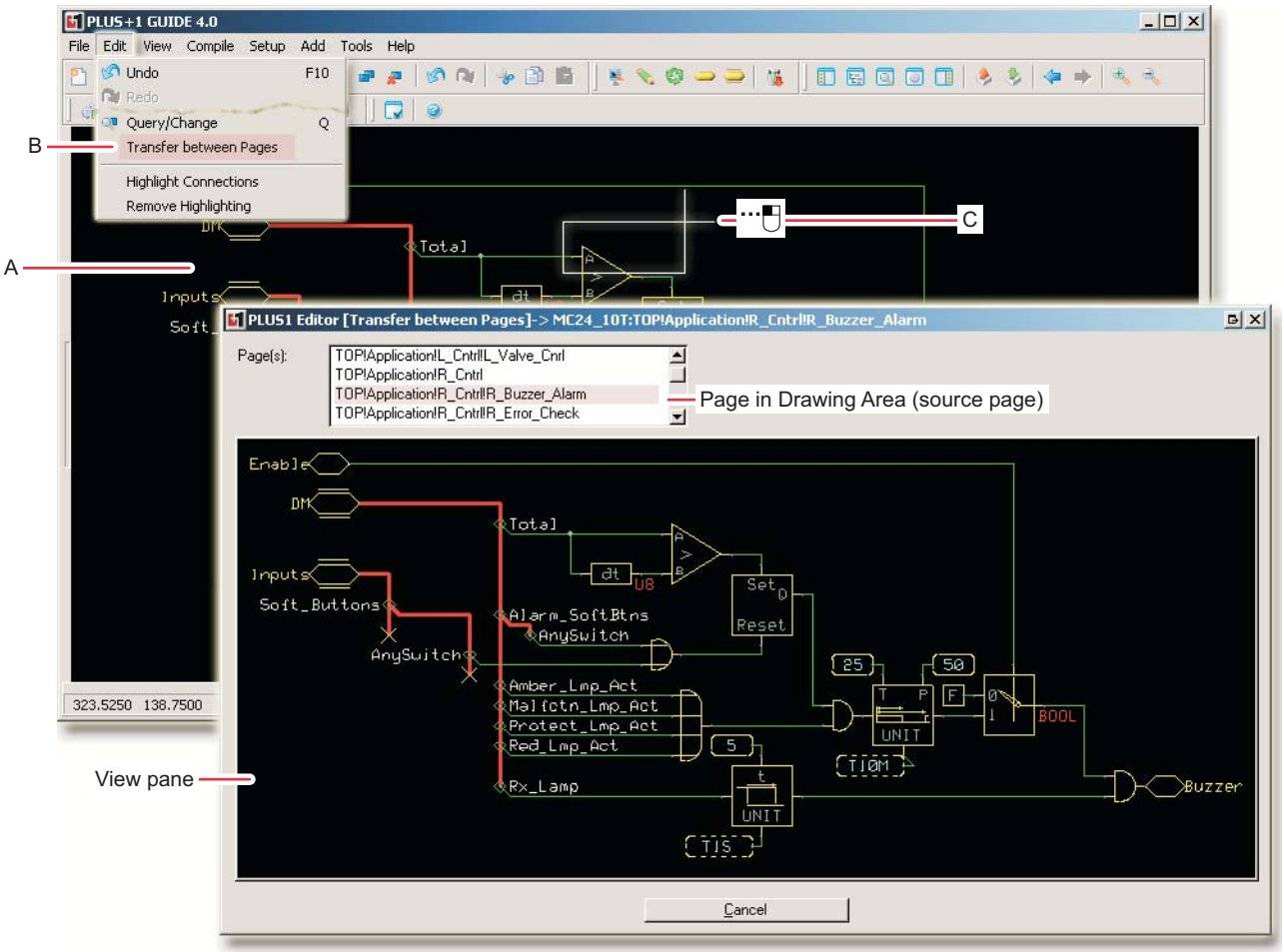
   - 星印(*) は削除される項目のクラスを特定します。
   - ダッシュ(–)は削除されない項目のクラスを特定します。

4. **OK** をクリックして項目を削除します。

5. F4 を押して図面エリアの表示を更新します。

ページの全内容を他のページにコピー

Transfer between Pages コマンドを使用して元ページの全ての内容を送り先のページにコピーすることができます。送り先のページはアプリケーション内のどのページでも可能です。元ページを送り先ページにコピーした後は、送り先ページで作業を継続するか元ページに戻るかを選択することができます。



1. 元ページの内容をコピーします。

   A. **PLUS+1 GUIDE** ウィンドウの図面エリア内に元ページ（コピーしたい内容を含むページ）を表示します。

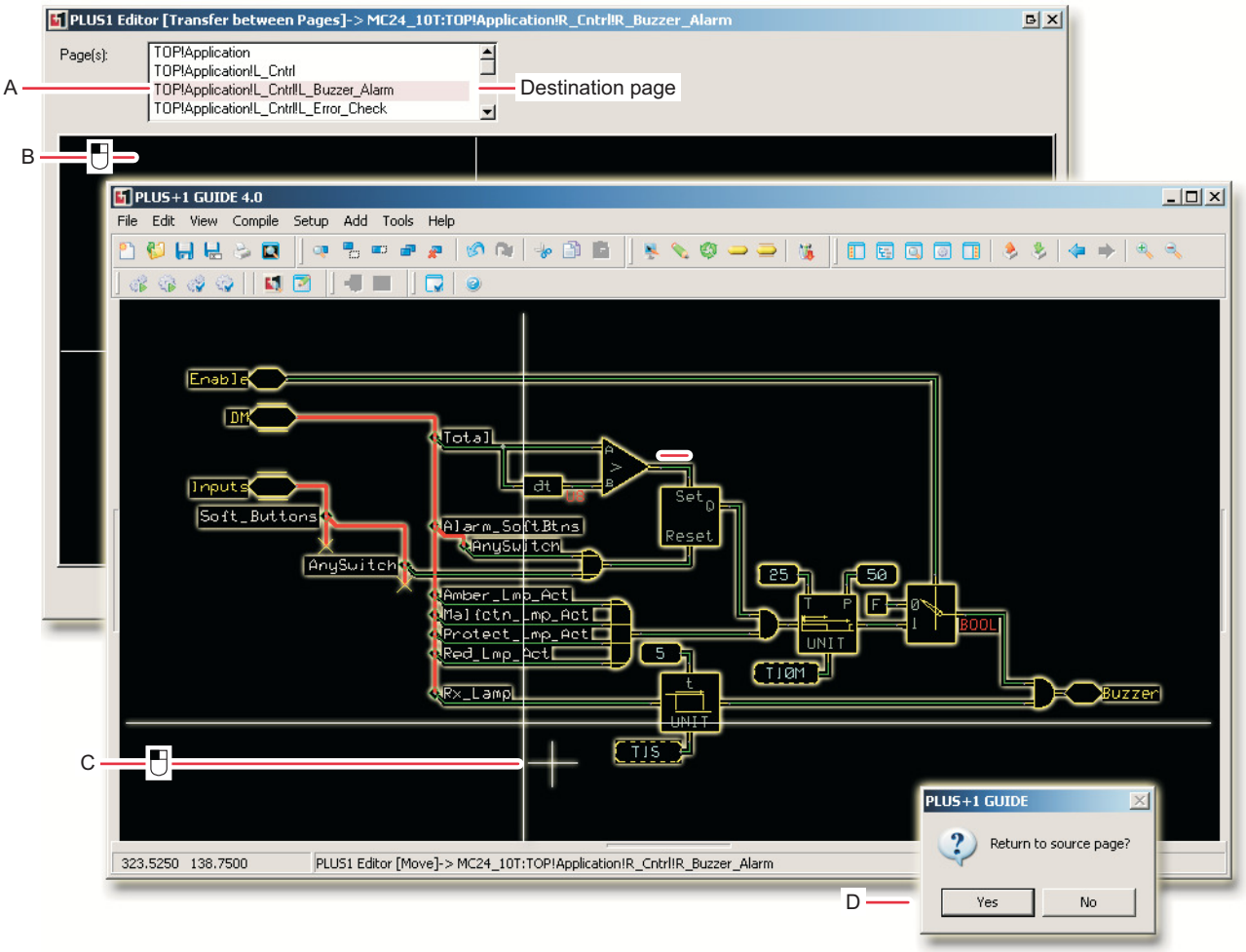
   B. **Edit** メニューで **Transfer between Pages** コマンドをクリックします。

   C. **PLUS+1 GUIDE** ウィンドウの図面エリア内でアイテム上をクリック＆ドラッグします。

   PLUS1 Editor [Transfer between Pages]ウィンドウが開いて以下のように表示します

   – **PLUS+1 GUIDE** ウィンドウの図面エリアの内容をビューペインに表示

   – アプリケーション内の全てのページを **Page(s)**にリスト



2. 元ページの内容を送り先ページにコピーします。

   A. **PLUS1 Editor [Transfer between Pages]**ウィンドウの **Page(s)**リストから送り先ページをクリックして選択します。

      PLUS1 Editor [Transfer between Pages]ウィンドウ内のビューペインに送り先ページが表示されます。

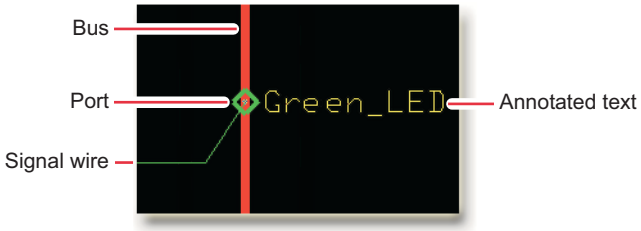
   B. **PLUS1 Editor [Transfer between Pages]**ウィンドウのビューペインをクリックしてこのウィンドウを閉じ、**PLUS+1 GUIDE** ウィンドウの図面エリアに送り先ページを表示します。

   C. **PLUS+1 GUIDE** ウィンドウの図面エリア内をクリックし、元ページからコピーした内容を送り先ページに配置します。

      PLUS+1 GUIDE ウィンドウは **Return to source page?**メッセージを表示します。

   D. **Yes** をクリックすると元ページに戻ります。**No** をクリックすると送り先ページで作業を継続できます。

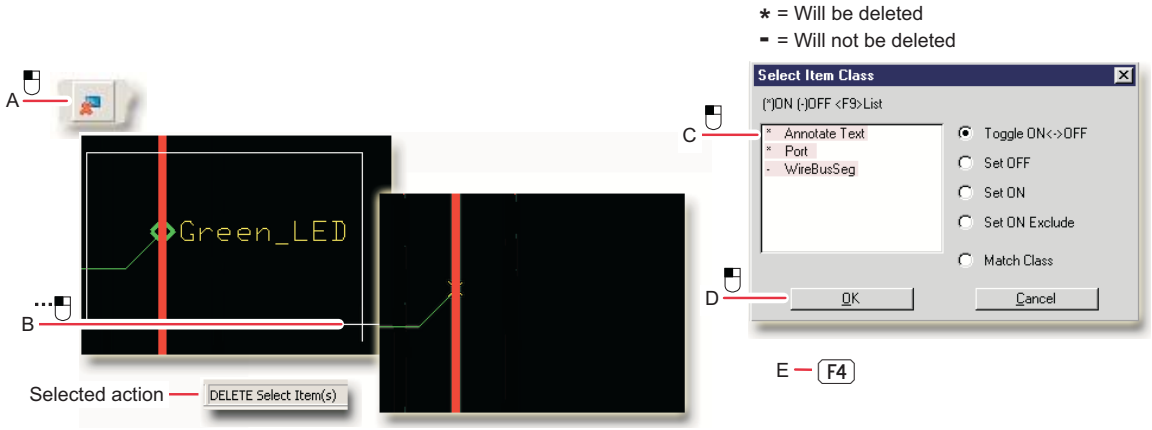
信号ーバス接続の削除



信号ーバス接続は以下の要素から成っています。

- 緑色の信号線
- 信号線が搬送した信号を特定する注釈文字列
- 信号線がバスに入るポート
- 赤色の信号バス

一般に信号ーバス接続を削除するのは、バスはそのままにしてワイヤ、注釈文字列、ポートを削除したいときです。



1. バスは削除せずに信号ーバス接続を削除します。

   A. ツールバーの **Delete** ボタンをクリックします。

   ステータスバーに削除を選択したことが表示されます。

   B. 図のようにクリック＆ドラッグで信号接続項目を選択します。

   マウスを放すと **Select Item Class** ウィンドウが開いて、選択項目をクラスで特定します。

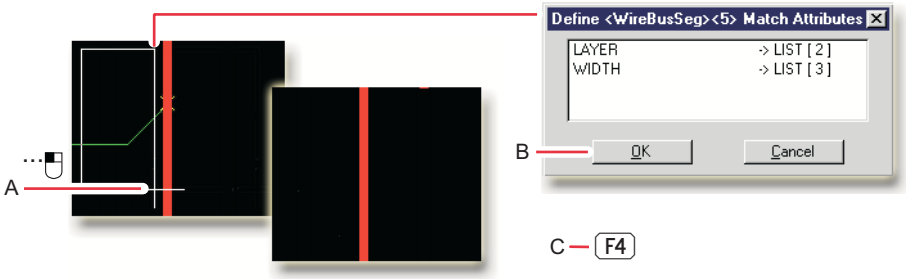
C. **Select Item Class** ウインドウで、**Toggle ON<->OFF** をクリックして、

- **Annotate Text** の横にある星印(*)
- **Port** の横にある星印(*)
- **WireBusSeg** の横にあるダッシュ(–)

をクリックします。

D. **OK** をクリックして選択した項目を削除します。

E. F4 を押して、図面エリアビューをリフレッシュします。



2. 残りのワイヤセグメントを削除します。

A. 図のようにクリック＆ドラッグでワイヤセグメントを選択します。

マウスを放すと **Attributes** ウィンドウが開いて、選択したワイヤの属性を示します。

B. **OK** をクリックして選択した項目を削除します。

C. F4 を押して、図面エリアビューをリフレッシュします。

この手順を繰返して、残りのワイヤセグメントを削除します。

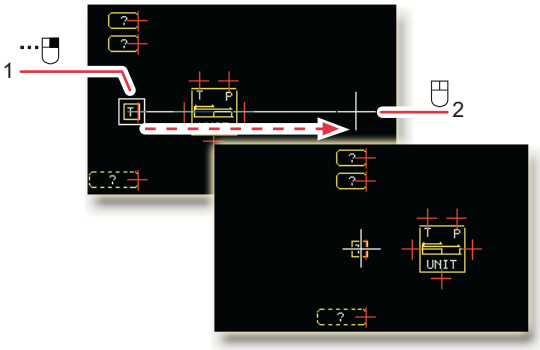
表示の更新

F4 を押して図面エリアの表示を更新します。

ページ全体の表示

Home を押してページ全体を図面エリアに合わせます。薄い青線(—)はページ境界を示します。

右クリック＆ドラッグによる表示のパン



1. 右クリックして、ページを移動したい方向にポインターを真直ぐにドラッグします。

   右クリックしたところに正方形が現れます。

2. マウスを放します。

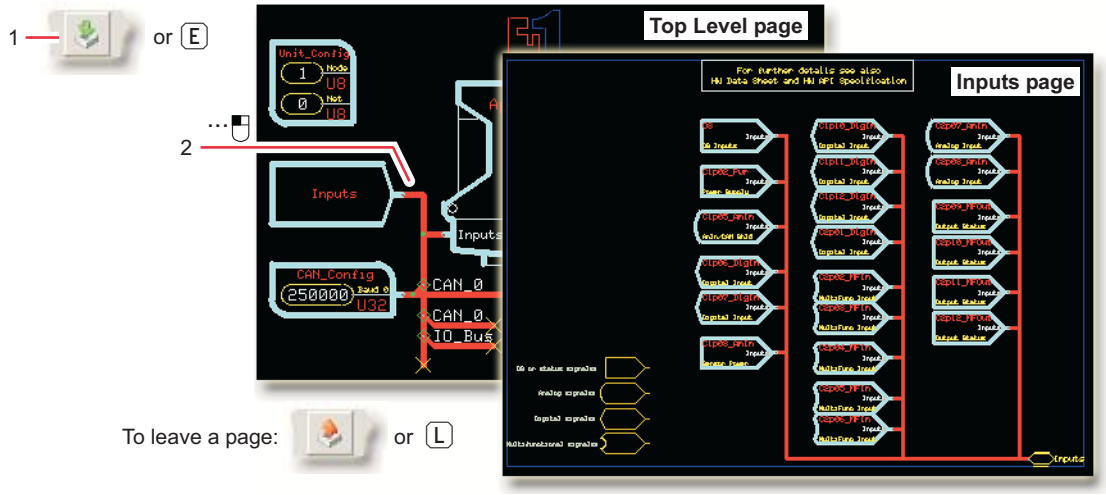
   正方形の下にあったエリアが、ポインター上に中心を移します。

右クリックによる表示のパン

ページを移動したい方向と反対の方向でマウスを右クリックします。

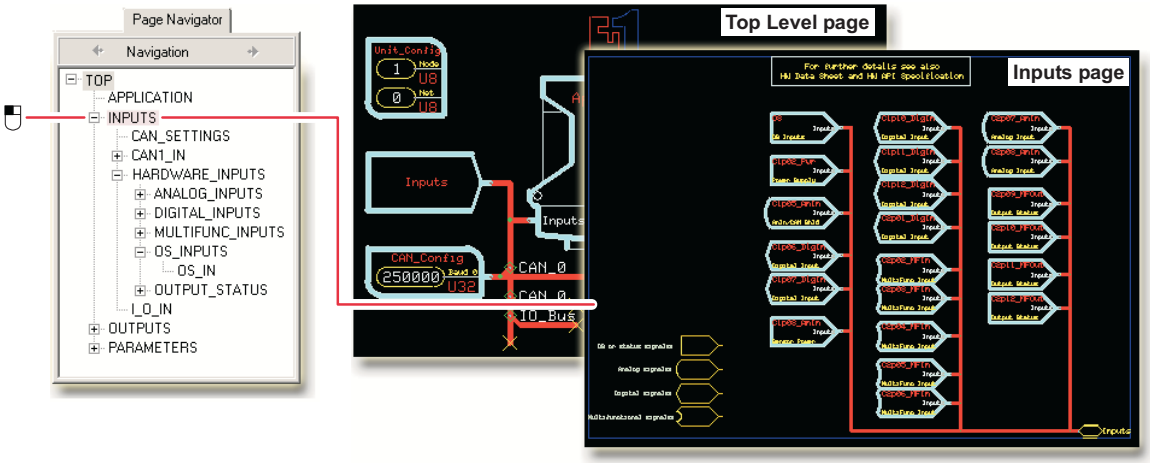
右クリックする位置がページの中心から遠いほど、ページが大きく移動します。

ボタンによるアプリケーションのナビゲーション



- ページに入るときは：
  1. ツールバーの **Enter Page** ボタンをクリックするか、E を押します。
  2. ページポートでドラッグするか、ページ境界内部をクリックします。
- ページを閉じるときは、ツールバーの **Leave Page** ボタンをクリックするか、L を押します。

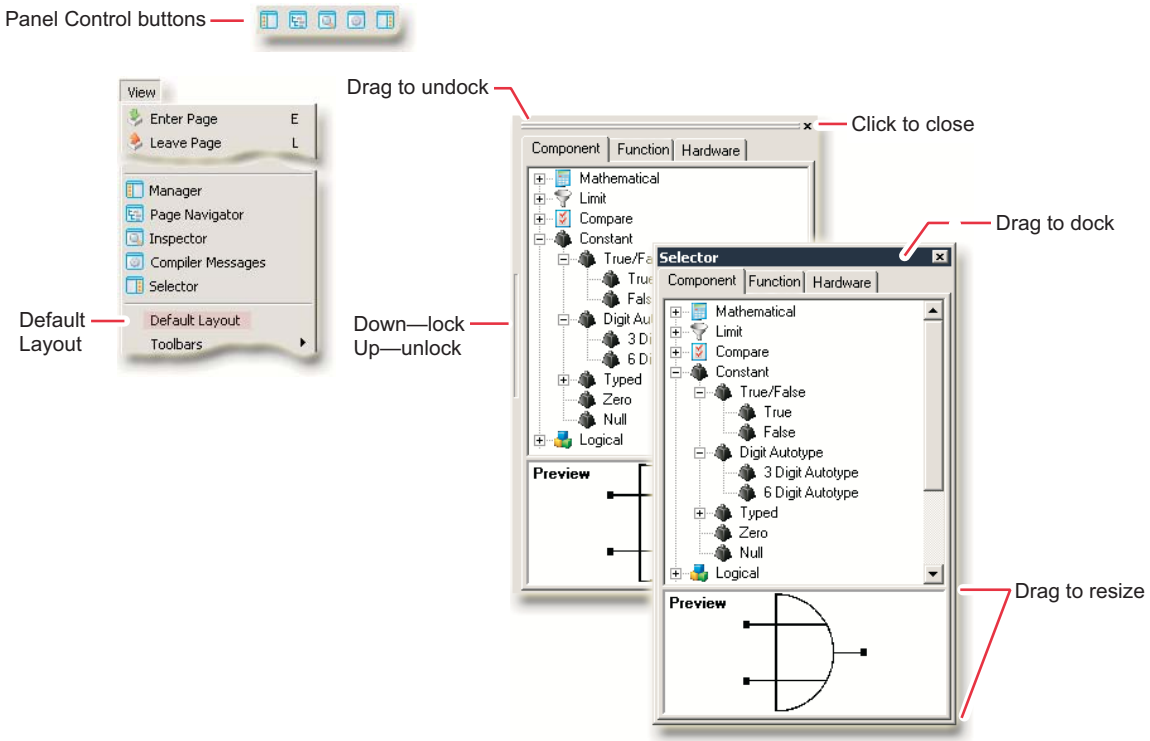
Page Navigatorタブによるアプリケーションのナビゲーション



- 直接ページを開くときは、**Page Navigator** タブのページをクリックします。

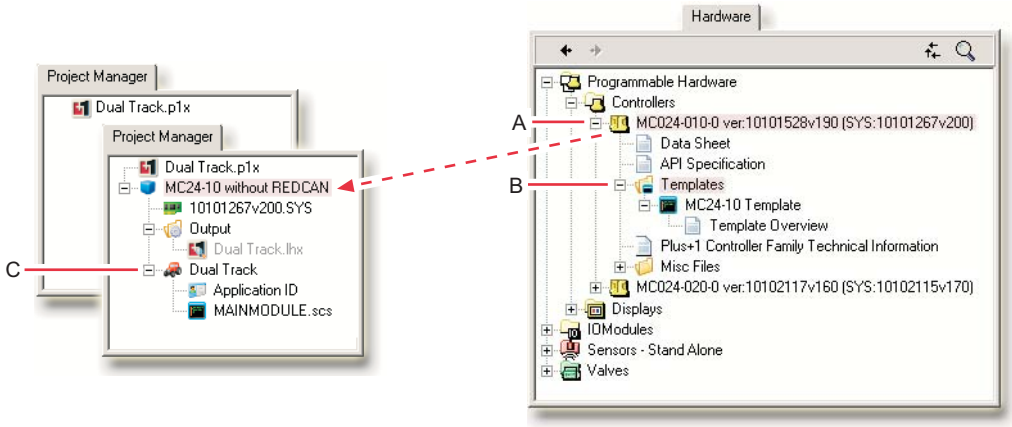
ペインの表示と非表示

PLUS+1 GUIDE ウィンドウでペインの表示と非表示を行うボタンは下図の通りです。（ペインによっては本図のボタンと異なる場合があります。）



- **View** メニューの **Default Layout** コマンドをクリックすると、**Manager**、**Inspector**、**Compiler**、**Selector** の各ペインをデフォルト位置にロックして表示します。
- **Default Layout** コマンドをクリックすると、ペインコントロールボタンまたは **View** メニューの **Manager**、**Page Navigator**、**Inspector**、**Compiler Messages**、**Selector** コマンドを使用してペインの表示と非表示を行うことができます。
- **PLUS+1 GUIDE** ウィンドウからペインを切り離すには、切り離しバーをドラッグしてペインをウィンドウから分離します。
- **PLUS+1 GUIDE** ウィンドウにペインを結合するには、ペインが結合するまでタイトルバーをウィンドウエッジ上にドラッグします。
- ペインを自動非表示（マウスの移動による表示と非表示）にするには、ペインコントロールボタンのロックを解除します。 ペインの縁にマウスを移動させたときだけペインを表示します。

ハードウェア記述のインストール

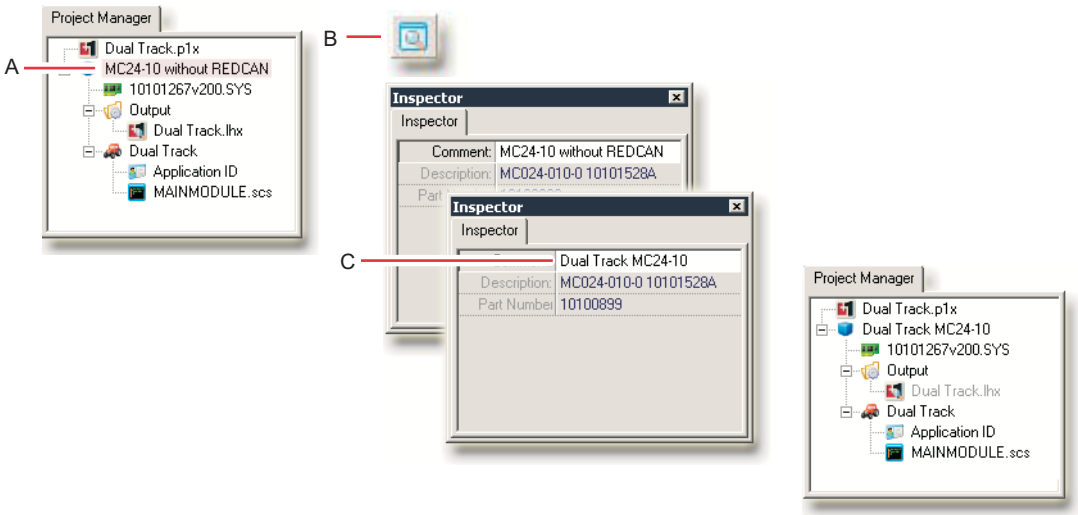


1. **Project Manager** タブにハードウェア記述をインストールします。

   A. ご使用の PLUS+1 ハードウェアモデルのハードウェア記述を探し、それを **Hardware** タブから **Project Manager** タブにドラッグします。

   B. **Project Manager** タブで、ハードウェア記述ツリーを全部開きます。

   C. **Hardware** タブでも、ハードウェア記述ツリーを全部開きます。

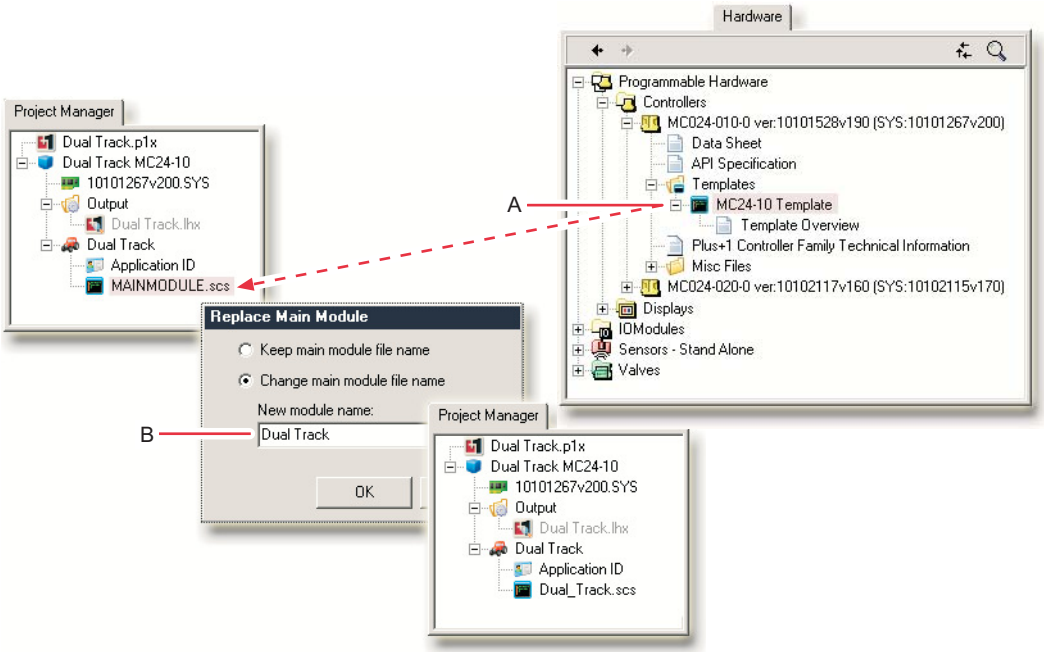


2. 必要に応じて、ハードウェアを説明するコメントを分かりやすいものに変更します。

   A. **Project Manager** タブで、ハードウェア記述コメントをクリックします。

   B. ツールバーの **Inspector** ボタンをクリックして、**Inspector** ペインを表示します。

   C. **Inspector** ペインの **Comment** 欄で、デフォルトのハードウェア記述コメントを変更します。



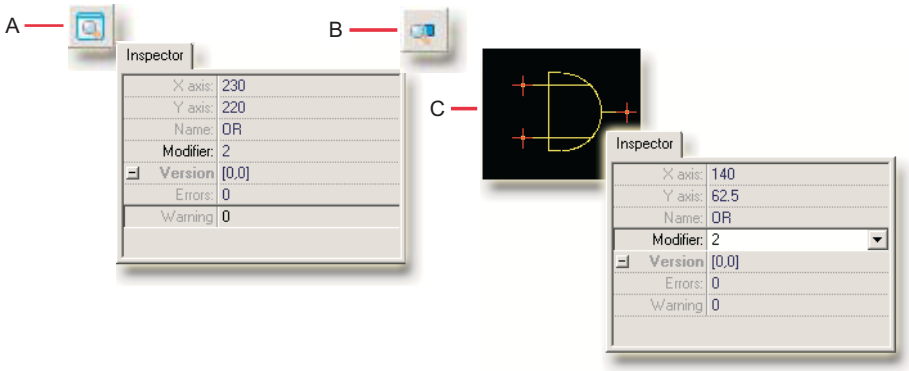
3. **Template** をインストールして名前を変更します。

   A. **Template** を **Hardware** タブから **Project Manager** タブにドラッグします。

   （**Template** を図面エリアにドラッグ＆ドロップすることもできます。）

   B. **Rename Main Module** ウィンドウを使用して、**Default** テンプレートに分かりやすい名前をつけます。

Inspectorタブによるプロパティの変更



1. **Inspector** タブにある項目のプロパティを表示します。

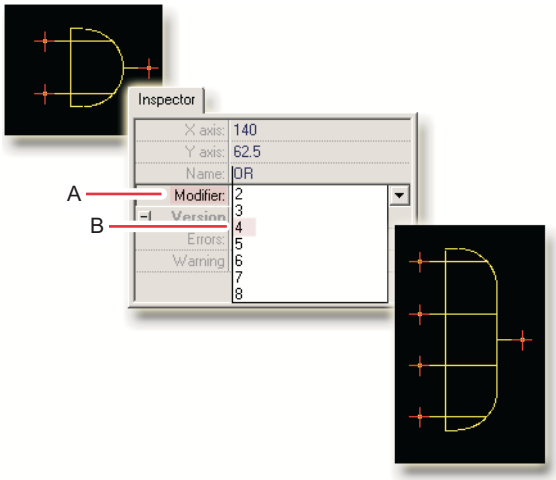
   A. ツールバーの **Inspector** ボタンをクリックします。

   Inspector タブが表示されます。

   B. ツールバーの **Query/Change** ボタンをクリックします。

   C. 図面エリアの項目をクリックします。

   Inspector タブがこの項目のプロパティを表示します。

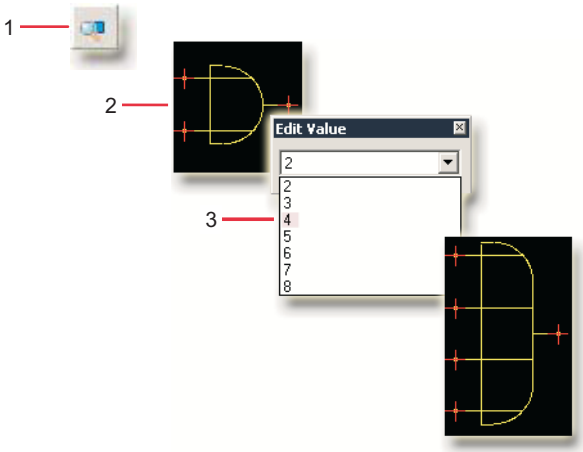


2. 項目のプロパティを変更します。

   A. **Inspector** タブで、変更したい項目をクリックします。

   B. 新規のプロパティ値を選択するか入力します。

ポップアップ編集ウィンドウによるプロパティの変更



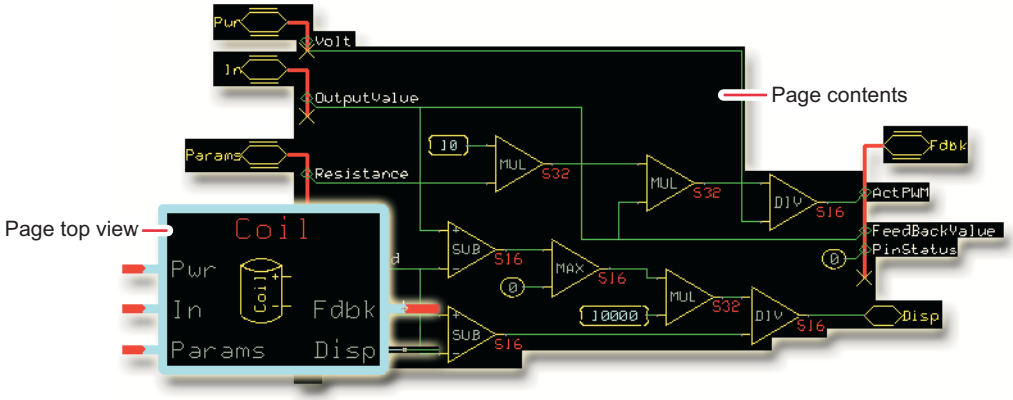
項目のプロパティを変更します。

1. ツールバーの **Query/Change** ボタンをクリックします。
2. 図面エリアで変更したいプロパティをクリックします。

   Edit Value ウィンドウがポップアップ表示されます。
3. プロパティ値を変更します。

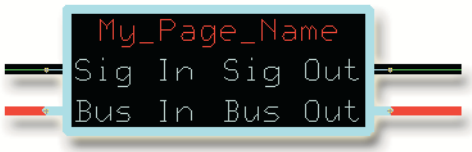
デフォルトでは、PLUS+1 GUIDE アプリケーションはポップアップ **Edit Value** ウィンドウが有効になっています。**Options** ウィンドウで、このウィンドウの有効、無効を設定します。
このウィンドウへのパスは、**Setup** メニュー> **Options** > **Options** ウィンドウ > **General** です。

ページの新規作成



ページのトップビューはページ内容の概観のことです。ページに入る前に見ることができます。

作成したページの内容によってページに必要とされる入力や出力の数が決まります。この手順では、ページを作成することに主眼を置き、ページの内容を追加することは述べません。



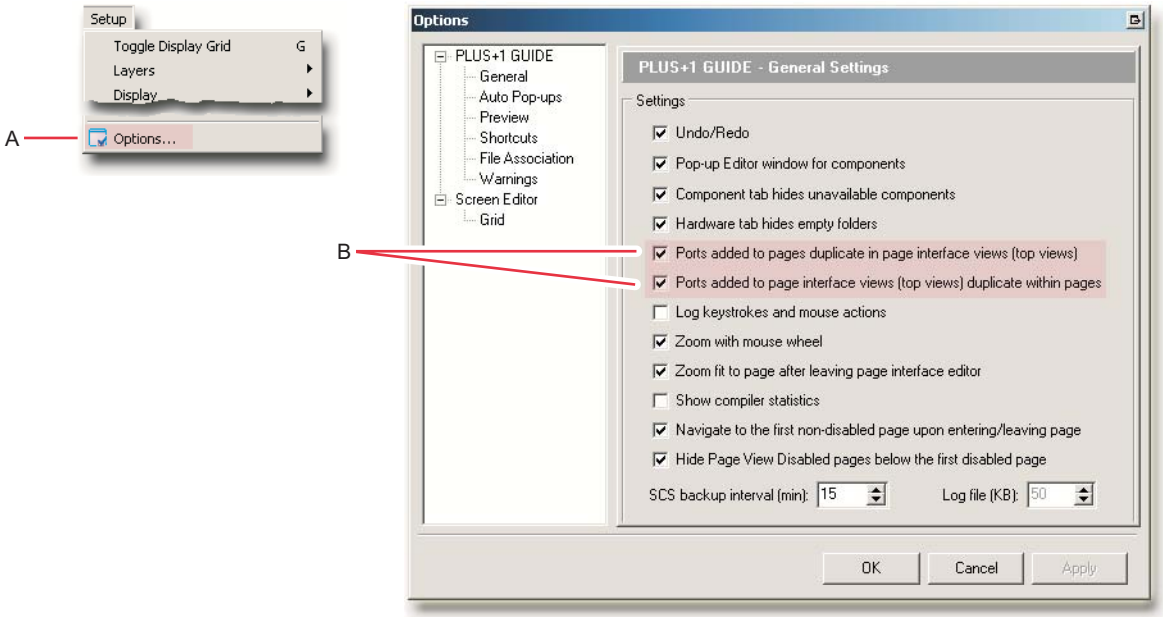
以下の手順では、前の図で示したページの「トップビュー」を作成します。このページには信号入力が一つ、バス入力が一つ、信号出力が一つ、バス出力が一つあります。

以下の手順では、前の図で示したページの「トップビュー」を作成します。このページには信号入力が一つ、バス入力が一つ、信号出力が一つ、バス出力が一つあります。

詳しい情報については、***94*** ページの ***About Pages, Page Top Views, and the Page Interface Editor Window*** を参照してください。

また、ページ要素のレイアウトやポートのラベル付けに関しては ***567*** ページの ***Page Layout Guidelines*** を参照してください。

@PAGENAME プレースホルダの **Layer** 値は **PageName** のままにしておいてください。この値を変更するとページ名が消失することがあります。また、コンパイルエラーや PLUS+1 プログラムの応用機能に支障をきたす場合があります。



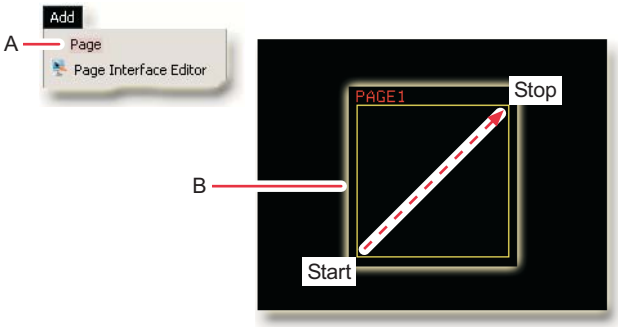
1. **Options** ページの設定をチェックします。

   A. PLUS+1 GUIDE ウィンドウの **Setup** メニューで、**Options** コマンドをクリックして **Options** ウィンドウを表示します。

   B. **Options** ウィンドウ内で以下のことを確認してください

   – **Ports added to pages duplicate in page interface views (top views)** オプションがチェックされていること

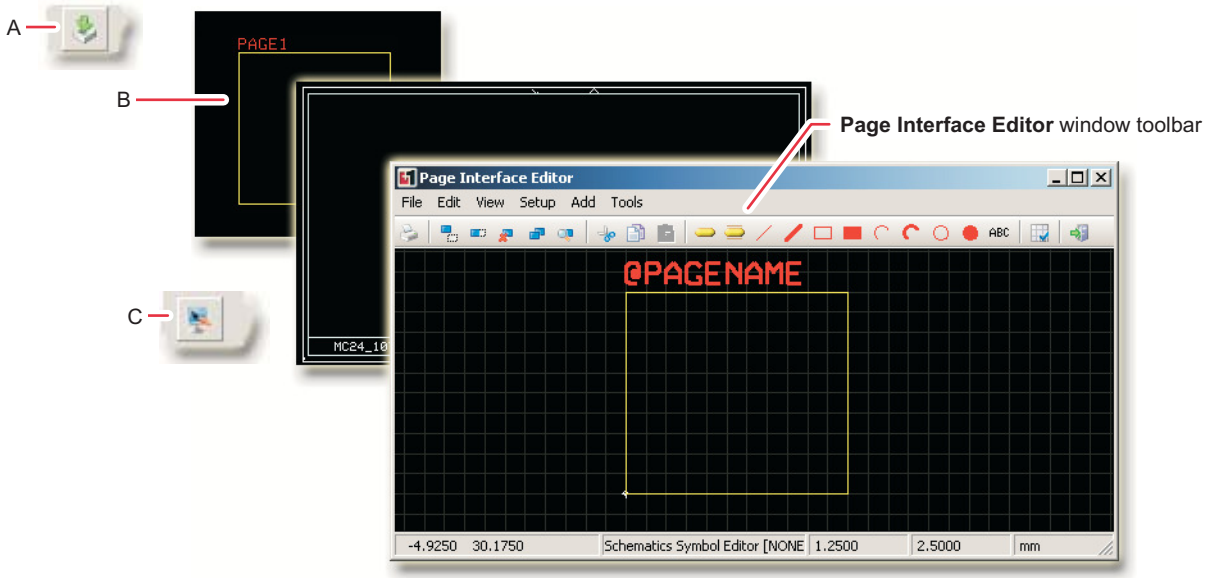
   – **Ports added to page interface views (top views) duplicate within pages** オプションがチェックされていること



2. ページを追加します。

   A. PLUS+1 GUIDE ウィンドウの **Add** メニューで **Page** コマンドをクリックします。

   B. PLUS+1 GUIDE ウィンドウの図面エリアで、左下から右上までクリック＆ドラッグしてページを追加します。



3. **Page Interface Editor** ウィンドウを開きます。

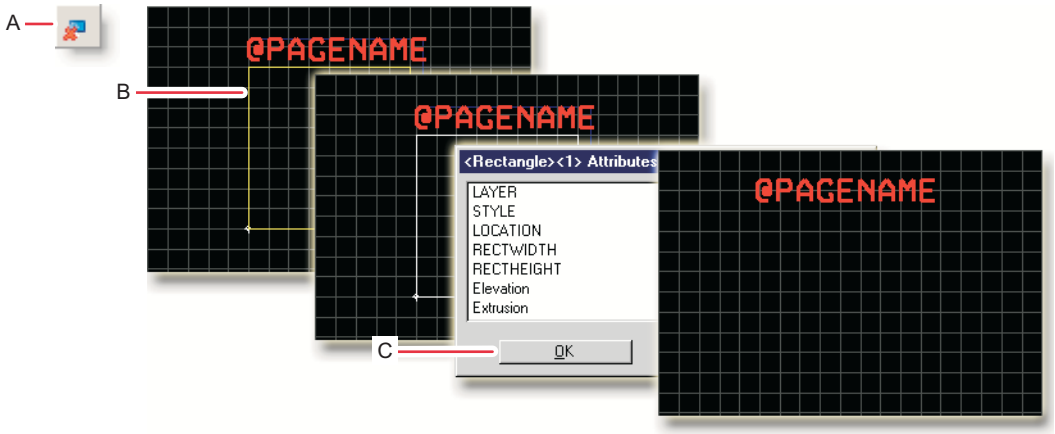
   Page Interface Editor ウィンドウでトップビューの外観を設計します。

   A. **PLUS+1 GUIDE** ウィンドウのツールバーで、**Enter Page** ボタンをクリックします。

   B. ページの境界をクリック&ドラッグしてページに入ります。

   C. **PLUS+1 GUIDE** ウィンドウのツールバーで、Page Interface Editor ボタンをクリックして、**Page Interface Editor** ウィンドウに入ります。

      Page Interface Editor ウィンドウが図面エリア内にページのトップビューを表示します。**Page Interface Editor** ウィンドウにあるツールバーのボタンを使ってページのトップビューを作成します。

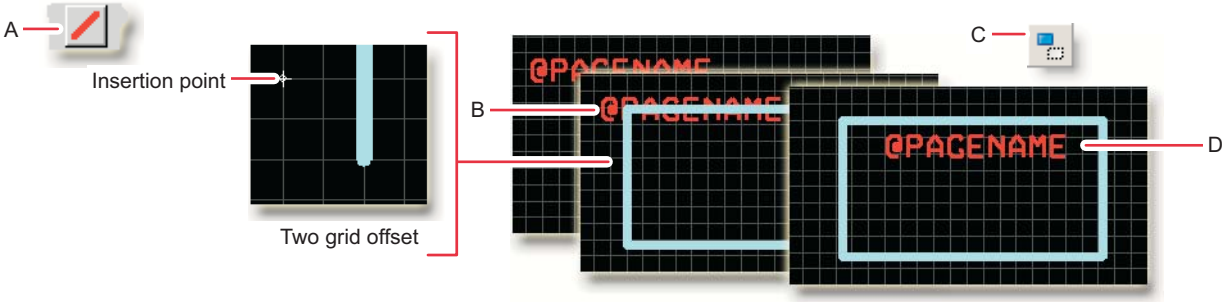


4. **Page Interface Editor** ウィンドウでページの長方形の境界線を削除します。

   A. **Page Interface Editor** ウィンドウのツールバーで **Delete** ボタンをクリックします。

   B. ページ境界上でドラッグします。ドラッグするとページの線が白色になって **Attributes**（属性）ウィンドウが開きます。

   C. **<Rectangle> Attributes** ウィンドウで **OK** をクリックします。



5. **Page Interface Editor** ウィンドウで、新たに太い青線でページの境界線を引いて、**@PAGENAME** の位置を変更します。

   （このとき、**@PAGENAME** は削除しないでください。@PAGENAME はステップ 16 で分かりやすいページ名を追加するときの仮の文字列です。）

   A. **Page Interface Editor** ウィンドウのツールバーで、太線ボタンをクリックします。

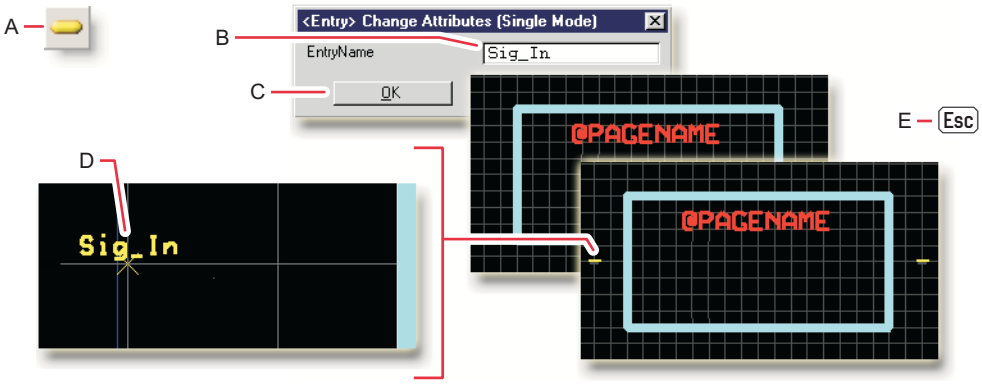
   B. クリック＆ドラッグして境界線を引きます。

      左の縦境界線の位置は、挿入点から右にグリッド線 2 本分（5.0mm）、下にグリッド線 2 本分離します。

      ページ境界線の始点と終点は必ずグリッドの交差点にしてください。

   C. **Page Interface Editor** ウィンドウのツールバーで、**Move** ボタンをクリックします。

   D. **@PAGENAME** をクリックして、ページ上辺のすぐ下に移動します。



6. ページインタフェースエディタビューで、必要に応じてワイヤポートを追加します。

   ワイヤポートは信号をページに引き込みます。

   バスポートを追加するときは、次のステップに進んでください。

   A. **Page Interface Editor** ウィンドウのツールバーで、**Entry** ボタンをクリックします。

      <Entry> Change Attributes (Single Mode)ウィンドウが開きます。

      （上図に示したボタンです。Entry Bus ボタンと間違えないように注意してください。）

   B. **<Entry> Change Attributes (Single Mode)**ウィンドウの **EntryName** にワイヤポートの名前を入力します。

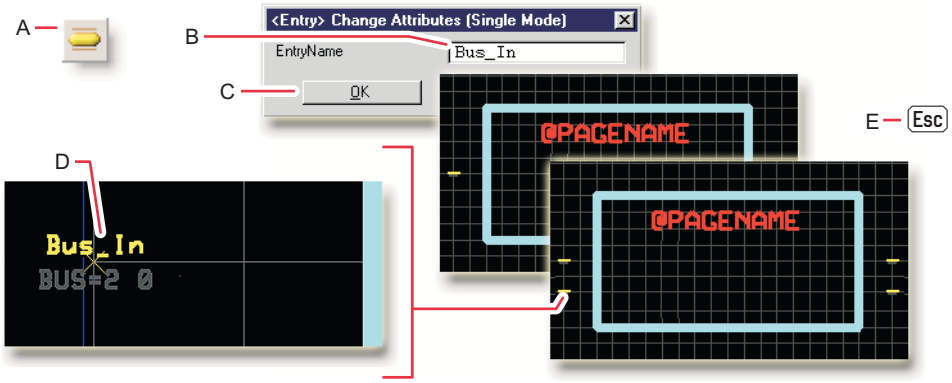
   C. **<Entry> Change Attributes (Single Mode)**ウィンドウの **OK** をクリックします。

   D. クリックしてワイヤポートを配置します。

      ポートはページ境界線から 2 グリッド離します。

   E. Esc を押します。

   ワイヤポートをさらに追加するときは、A～E のステップを繰返します。



7. **Page Interface Editor** ウィンドウで、必要に応じてバスポートを追加します。

   バスポートはバスをページ境界線に引き込みます。

   A. ツールバーの **Entry Bus** ボタンをクリックします。

      （上図に示したボタンです。Entry ボタンと間違えやすいので注意してください。）

      <Entry> Change Attributes (Single Mode)ウィンドウが開きます。

   B. **<Entry> Change Attributes (Single Mode)**ウィンドウの **EntryName** にバスポートの名前を入力します。

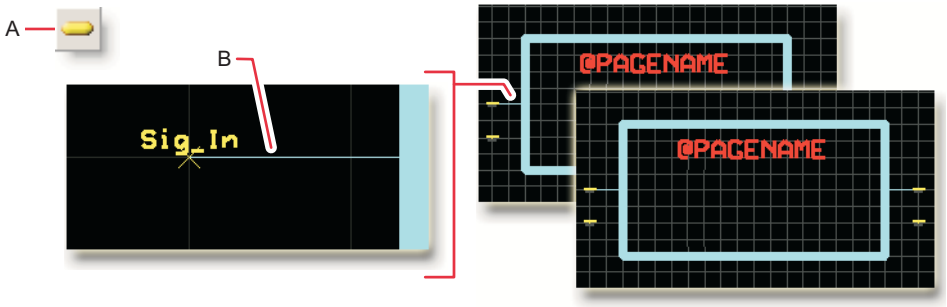
   C. **<Entry> Change Attributes (Single Mode)**ウィンドウの **OK** をクリックします。

   D. クリックしてバスポートを配置します。

      ポートはページ境界線から 2 グリッド離します。

   E. Esc を押します。

   バスポートをさらに追加するときは、A～E のステップを繰返します。

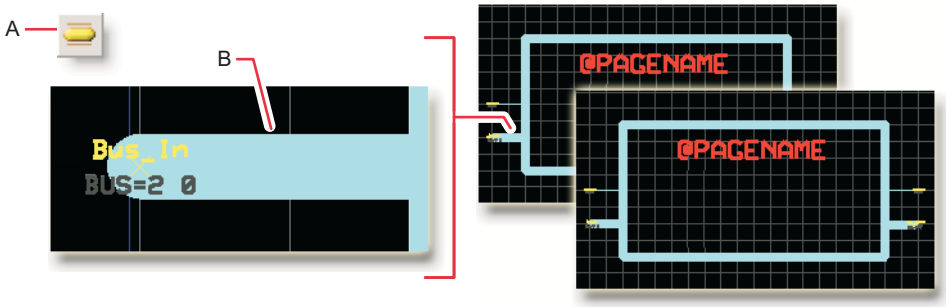


8. **Page Interface Editor** ウィンドウで、ワイヤポートとページ境界の間に細い線を引きます。

   この線に機能はありませんが、トップビューが見やすくなります。

   A. **Page Interface Editor** ウィンドウのツールバーで、Thin Line ボタンをクリックします。

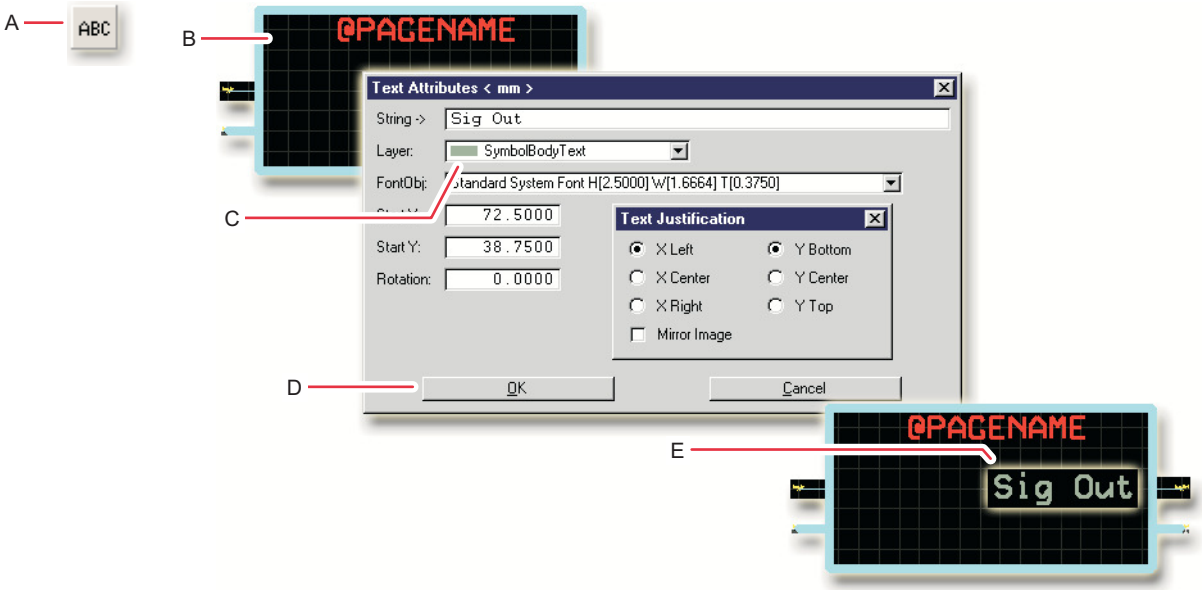
   B. 各ワイヤポートとページ境界線の間に細い線を引きます。



9. **Page Interface Editor** ウィンドウで、バスポートとページ境界の間に太い線を引きます。

   A. **Page Interface Editor** ウィンドウのツールバーで、Thick Line ボタンをクリックします。

   B. 各バスポートとページ境界線の間に太い線を引きます。



10. **Page Interface Editor** ウィンドウで、各エントリーに文字ラベルを追加します。

    A. **Page Interface Editor** ウィンドウのツールバーで、**Text** ボタンをクリックします。

    B. ページの任意の場所をクリックします。

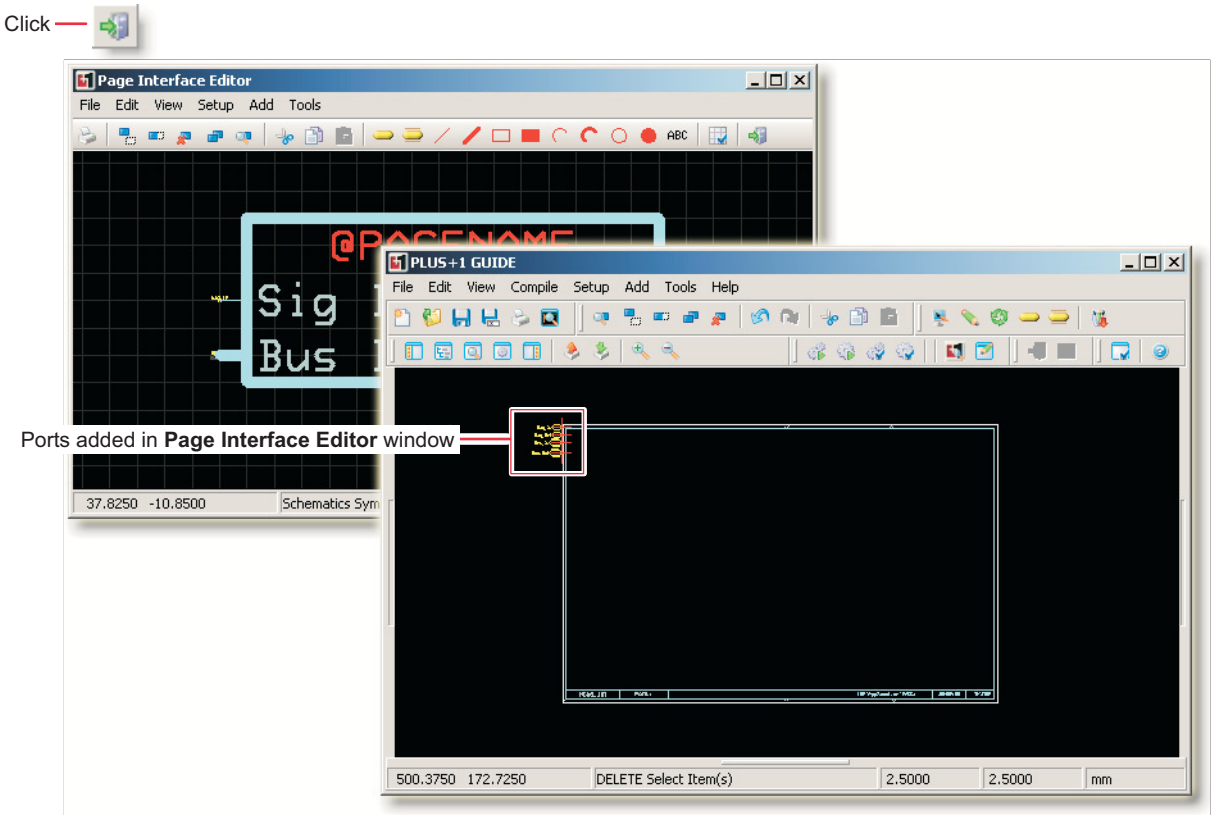
    Text Attributes ウィンドウが開きます。

    C. **Text Attributes** ウィンドウの **String**(文字列)欄に文字ラベル名を入力します。

    D. **OK** をクリックします。

    E. ラベルを配置したい場所をクリックします。

A～E のステップを繰返して、他の全てのエントリーに文字ラベルを追加します。



11. **Page Interface Editor** ウィンドウを閉じます。

    Page Interface Editor ウィンドウのツールバーで Close ボタンをクリックしてウィンドウを閉じて、**PLUS+1 GUIDE** ウィンドウに戻ります。

    PLUS+1 GUIDE ウィンドウの図面エリアに新しいページのページビューが表示されます。

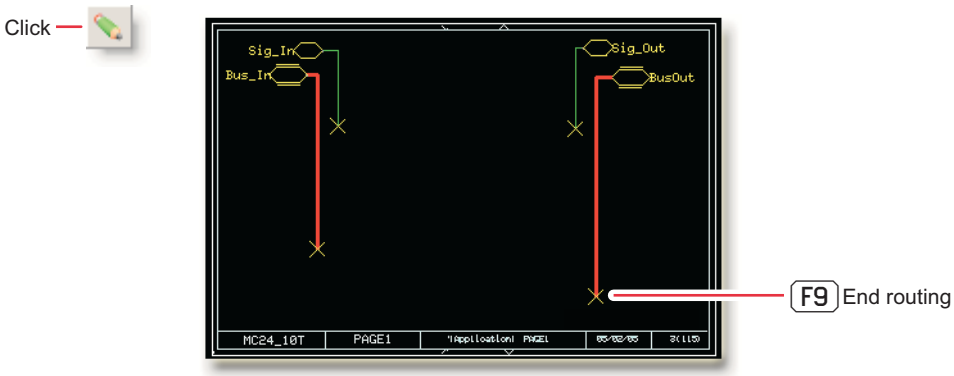
    Page Interface Editor ウィンドウで追加したポートがページの左上隅に現れます。

12. ポートをページ内に配置します。

PLUS+1 GUIDE ウィンドウのツールバーで Move ボタンをクリックします。

ポートをページ上でドラッグします。

F8 をクリックするとポートが左右反転します。

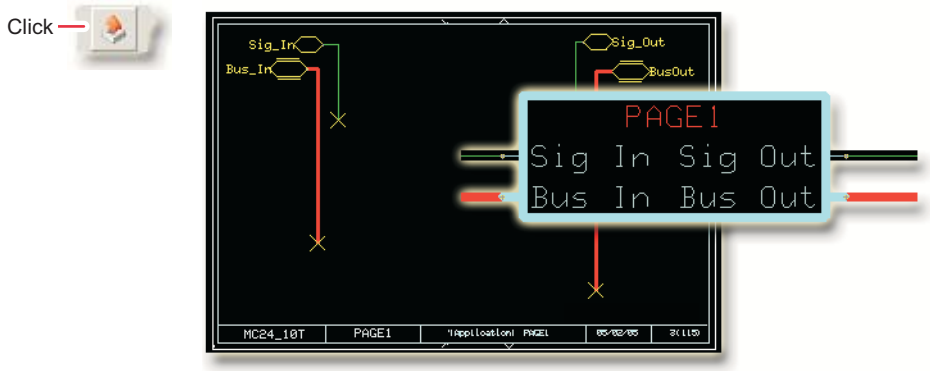


13. ページ上のポートにバスやワイヤを追加します。

PLUS+1 GUIDE ウィンドウのツールバーで Route ボタンをクリックします。

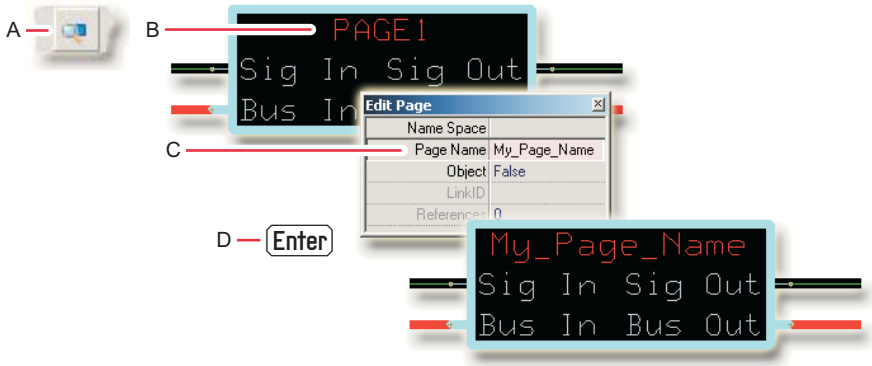
ポートをクリックし、ドラッグして配線を引きます。

F9 を押すとバスやワイヤの配線をどこにも接続しないまま終端します。



14. ウィンドウのツールバーで Leave Page ボタンをクリックしてページのトップビューに行きます。

（ページのトップビューは **Page Interface Editor** ウィンドウで作成済みです。）



15. ページに分かりやすい名前をつけます。

A. **PLUS+1 GUIDE** ウィンドウのツールバーで、Query/Change ボタンをクリックします。

B. ページ名（ここでは **Page1**）をクリックします。

Edit Page ウィンドウが開きます。

C. **Edit Page** ウィンドウで、分かりやすいページ名を入力します。

D. Enterを押します。.

View Logical Netの使用

View Logical Net ツールは選択したルートの全ての接続を複数のページに渡ってトレースします。

View Logical Net ツールは以下のような用途に使われます

- 選択したルートの全ての接続をトレースします。

  トレースをすることで、選択したルートが何をしているか、あるいはルートを変更すると何が起こるかを理解するのに役立ちます。

- 以下のようなコンパイラエラーの原因となる接続間違いあるいは接続漏れを見つけることができます。
  - *** ERROR *** MULTIDEFINED NET.

    典型例として、出力を二つ接続するとこのエラーが発生します。
  - *** ERROR 21 *** NO OUTPUT CONNECTED ON ENTRY.

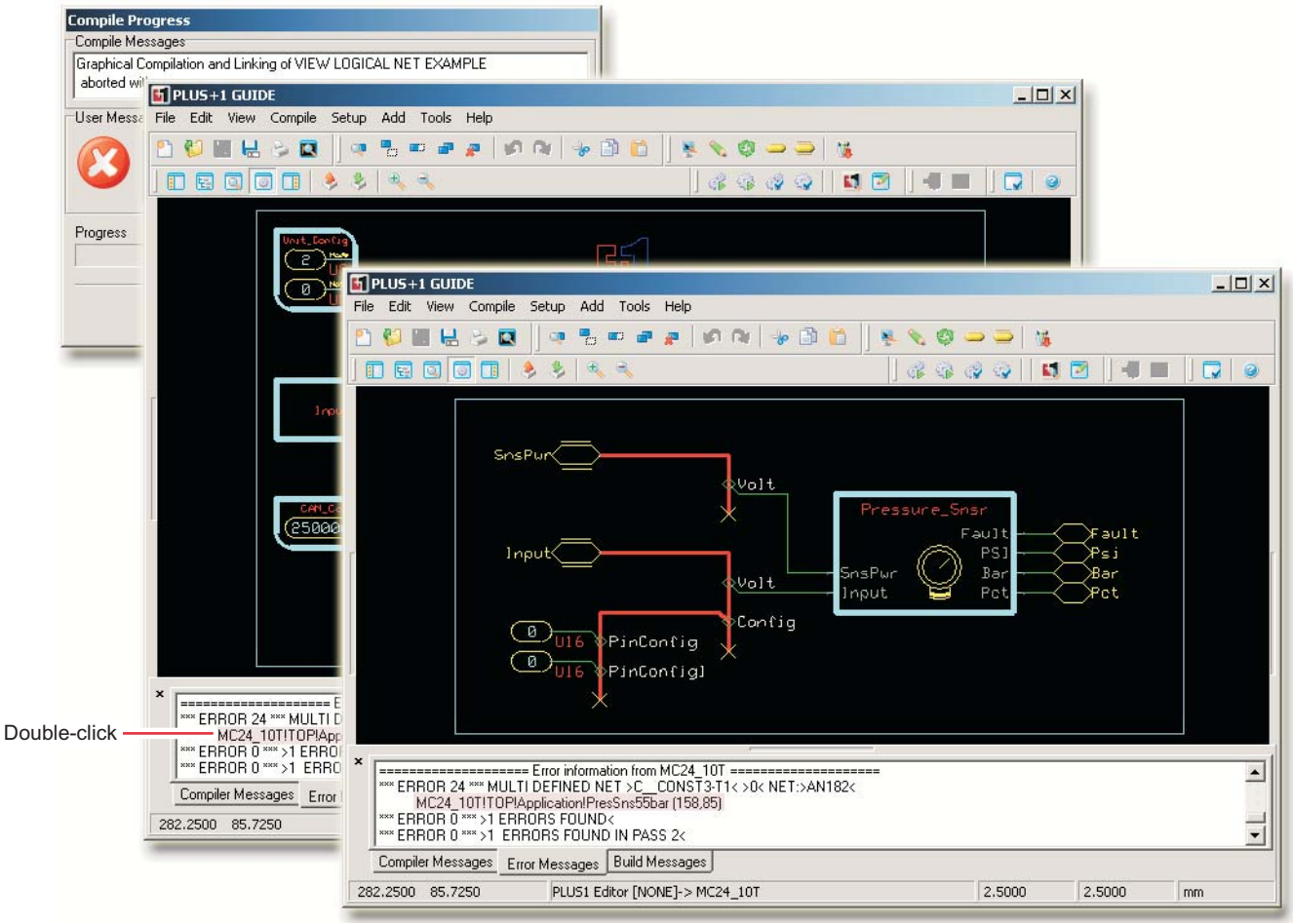
    典型例として、コンポーネントに入力を接続しないとこのエラーが発生します。
  - *** ERROR 15 *** NET TYPE REDEFINED.

    典型例として、CAN コンポーネントの出力を他のコンポーネントの出力に接続するとこのエラーが発生します。

View Logical Net ツールを理解する良い方法は、よく分かっているルートをトレースしてみることです。

以下の例はどのように **View Logical Net** ツールが MULTIDEFINED NET コンパイラエラーを見つけるかを示しています。

この例では、二つの **3 Digit Autotype** 定数コンポーネントが同じ **PinConfig** 入力に接続されていたのが原因で MULTIDEFINED NET コンパイラエラーが発生した場合を示します。

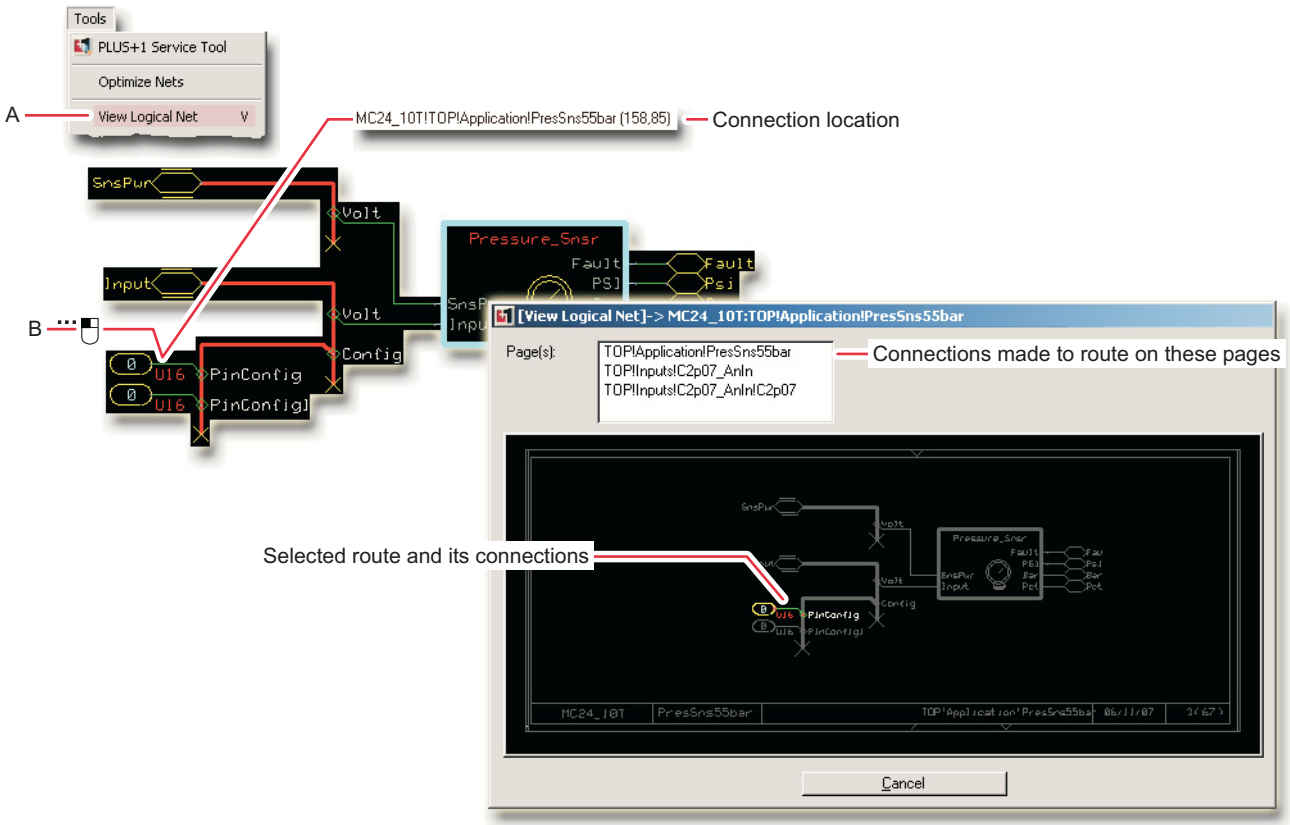


1. コンパイラがエラーを検出したページに行きます。

   Error Messages タブでエラーの位置を示す行をダブルクリックします。

   図面エリアにコンパイラがエラーを検出したページが表示されます。

   （コンパイラがエラーを検出したページとエラーの原因となったページはいつも同じとは限りません。）



2. 選択したルートに対して **View Logical Net** ウィンドウを表示します。

   A. **Tools** メニューで **View Logical Net** コマンドをクリックします。

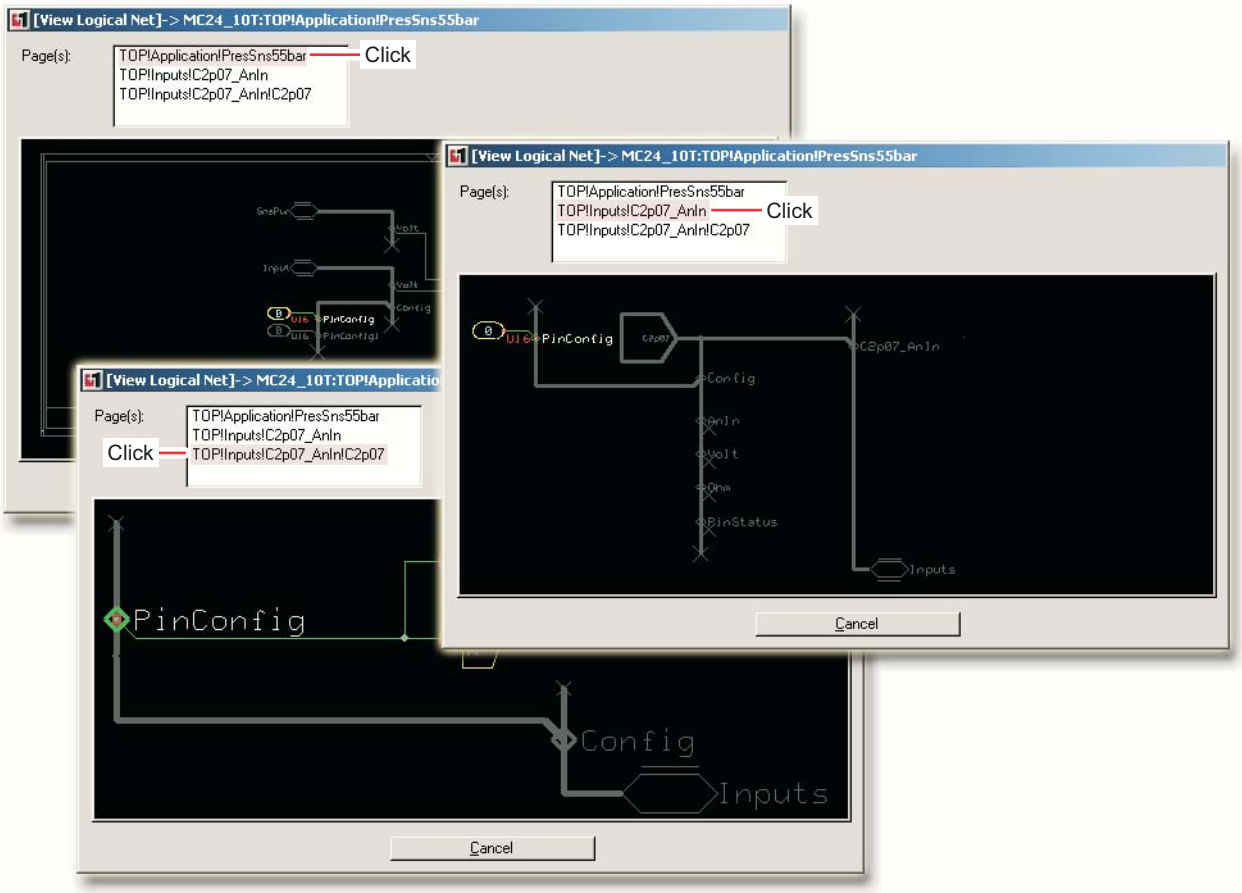
   B. エラーメッセージ内で特定されたルート接続点の位置を選択します。

   View Logical Net ウィンドウが開き、

   – Page Preview ペイン内でエラーメッセージ中で特定されたページが表示されます

   – 選択したルートとその接続がハイライトされ、その他の部分がグレー表示されます

   – **Page(s)**ペイン内で、選択したルートに接続されたページのリストが表示されます

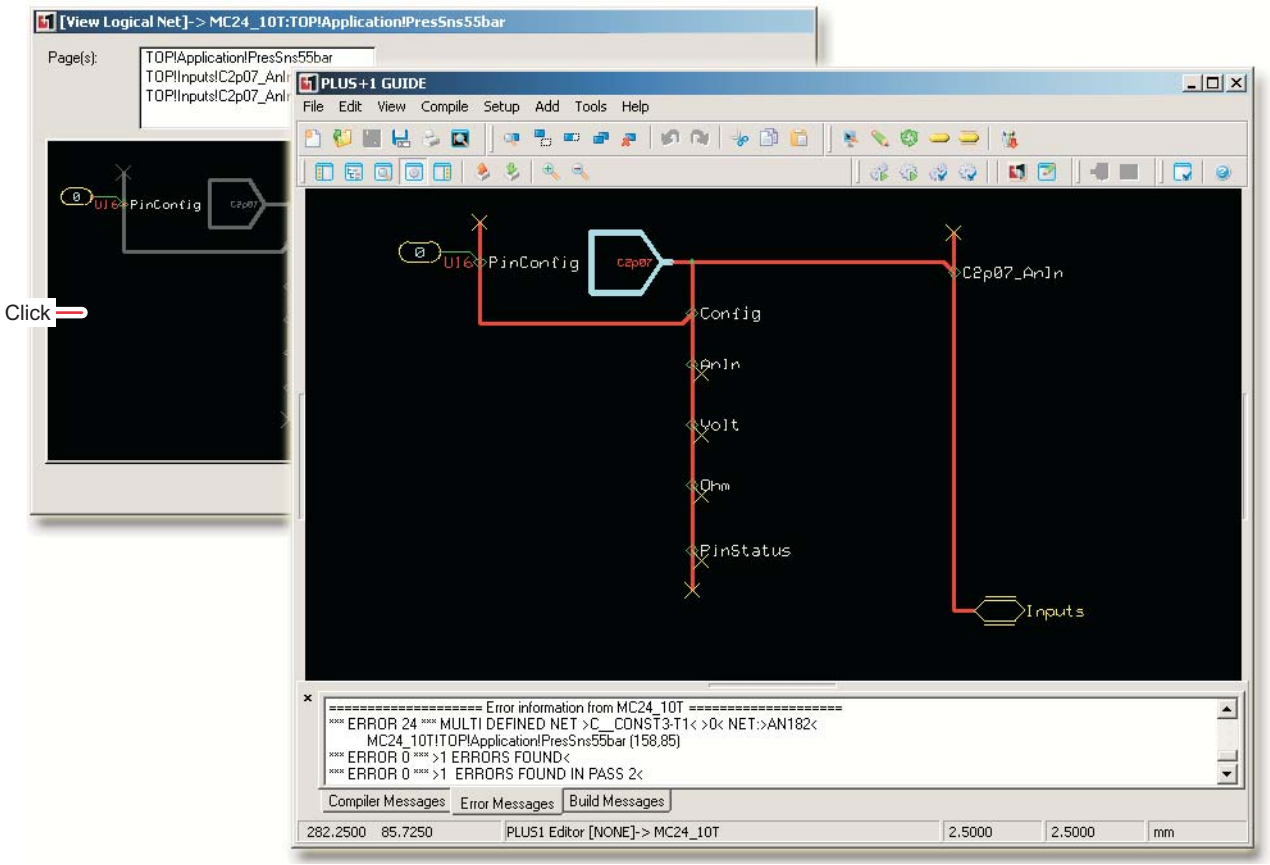


3. **View Logical Net** ウィンドウを使用してコンパイラエラーの原因を特定します。

   View Logical Net ウィンドウ内で **Page(s)**ペインにリストされているページをクリックして

   - 選択したルートをトレースします。
   - コンパイラエラーの原因を特定します。

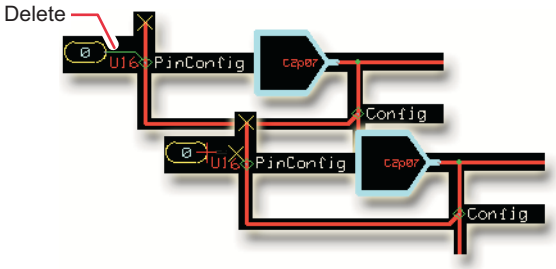
   View Logical Net ウィンドウ内でパンやズーム操作を行うこともできます。



4. 図面エリアに戻り、エラーの原因を修正します。

   View Logical Net ウィンドウ内でコンパイラエラーの原因となったページをクリックします。

   View Logical Net ウィンドウが閉じて図面エリアに原因のページが表示されます。



   コンパイラエラーの原因を修正します。

5. アプリケーションを再コンパイルします。

Read-only Parameters（読み取り専用パラメータ）ファイルの作成

読み取り専用パラメータファイルはアプリケーションの設定に使われる一連の値をダウンロード可能な lhx ファイルとしたものです。

読み取り専用パラメータファイルは、設定値を除いて共通のロジックにより制御される、例えば油圧モータファミリなどの一連のデバイスを制御する場合に有用です。

各々のデバイスに対して個別のアプリケーションを作成する代わりに、一連のデバイス群に共通のアプリケーションを個別に設定する各々異なる読み取り専用パラメータファイルを作成することができます。

- **444**ページの*読み取り専用パラメータファイルの新規作成*では読み取り専用パラメータファイルを新規に作成する方法を記述しています。読み取り専用パラメータファイルを持ったことのないアプリケーション用に読み取り専用パラメータファイルを作成する場合はこの手順に従ってください。
- **453**ページの*異なる値を持つ読み取り専用パラメータファイルの作成*では異なる値を持つ読み取り専用パラメータファイルを追加する方法を記述しています。既に読み取り専用パラメータファイルがあり、異なる値を持つファイルを追加したい場合はこの手順に従ってください。
- **457**ページの*パラメータを変更したアプリケーションのための新規読み取り専用パラメータファイルの作成*では、読み取り専用パラメータファイルを持つアプリケーションにおいてパラメータを変更した後で新たに読み取り専用パラメータファイルを作成する方法を記述しています。パラメータを変更（追加、削除、名前の変更、順序の変更）し、新しく読み取り専用パラメータファイルを作成したい場合にはこの手順に従ってください。

読み取り専用パラメータファイルの新規作成

この手順では読み取り専用パラメータファイルを新規に作成する方法を記述しています。読み取り専用パラメータファイルを持ったことのないアプリケーション用に読み取り専用パラメータファイルを作成する場合はこの手順に従ってください。

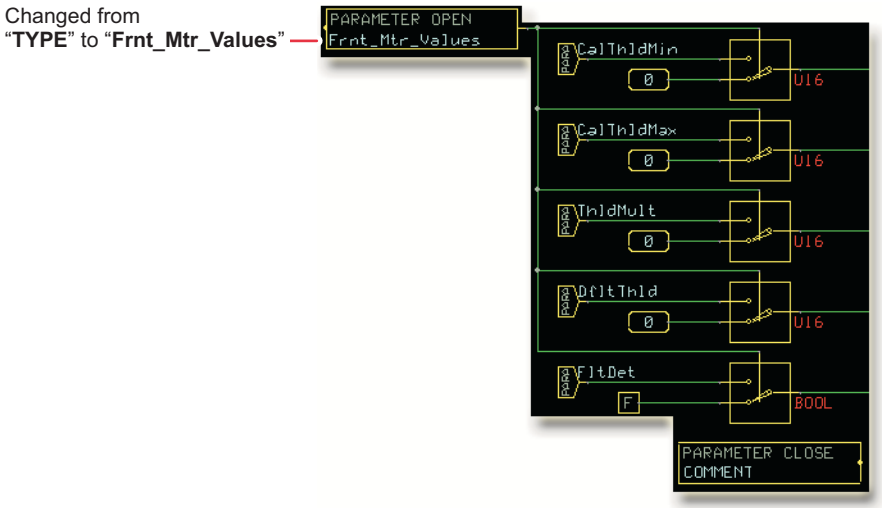
この例では以下の値を持つ読み取り専用パラメータファイルを作成します。

読み取り専用パラメータ

名前	信号の型	値
CalThldMin (Calibration Threshold Minimum)	U16	5400
CalThldMax (Calibration Threshold Maximum)	U16	8100
ThldMult (Threshold Multiplier)	U16	1000
DfltThld (Default Threshold)	U16	7500
FltDet (Fault Detect)	BOOL	1 (1 =T; 0 =F)

1. PLUS+1 GUIDE プログラムを開始します。
2. 読み取り専用パラメータ lhx ファイルを作成したい GUIDE プロジェクトをオープンします。

読み取り専用パラメータファイルを作成するのに使うコンポーネントについての詳細な情報は、***334*** ページの ***Read-only Parameter****(読取専用パラメータ)*を参照してください。

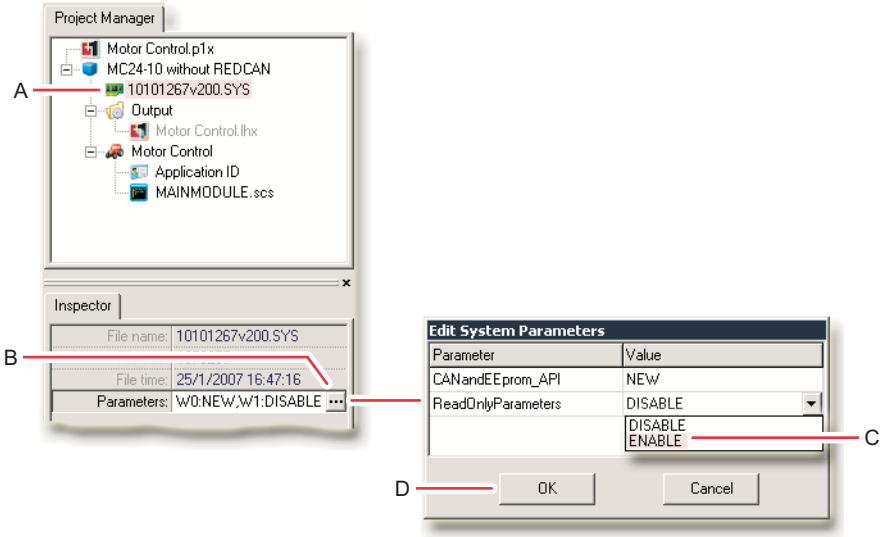


3. アプリケーションに読み取り専用パラメータを追加するのに必要なロジックを作成します。

   この例では、

   - **PARAMETER OPEN** コンポーネントは
     - ページ上のパラメータセットの開始点をマークします
     - **TYPE** 値が一致した読み取り専用パラメータ lhx ファイルをダウンロードしない限りブール値 F を出力します
     - **TYPE** 値が一致した読み取り専用パラメータ lhx ファイル（ここでは **Frnt_Mtr_Values**）をダウンロードするとブール値 T を出力します。
   - **Read-only Parameter Input** コンポーネントが読み取り専用 lhx ファイルにダウンロードされた値を保持します。
   - **Switch** コンポーネントが、
     - **Parameter Open** コンポーネントがブール値 F を出力しているときは 0 あるいは F の値をアプリケーションに出力します。
     - **Parameter Open** コンポーネントがブール値 T を出力しているときは **Read-only Parameter Input** コンポーネントからの値をアプリケーションに出力します。
   - **PARAMETER CLOSE** コンポーネントでページ上のパラメータセットの終点をマークします。

他のページのパラメータセットについて続けるには、同じ **Type** 値で **PARAMETER OPEN** でセットをオープンし、**Parameter Close** でクローズします。



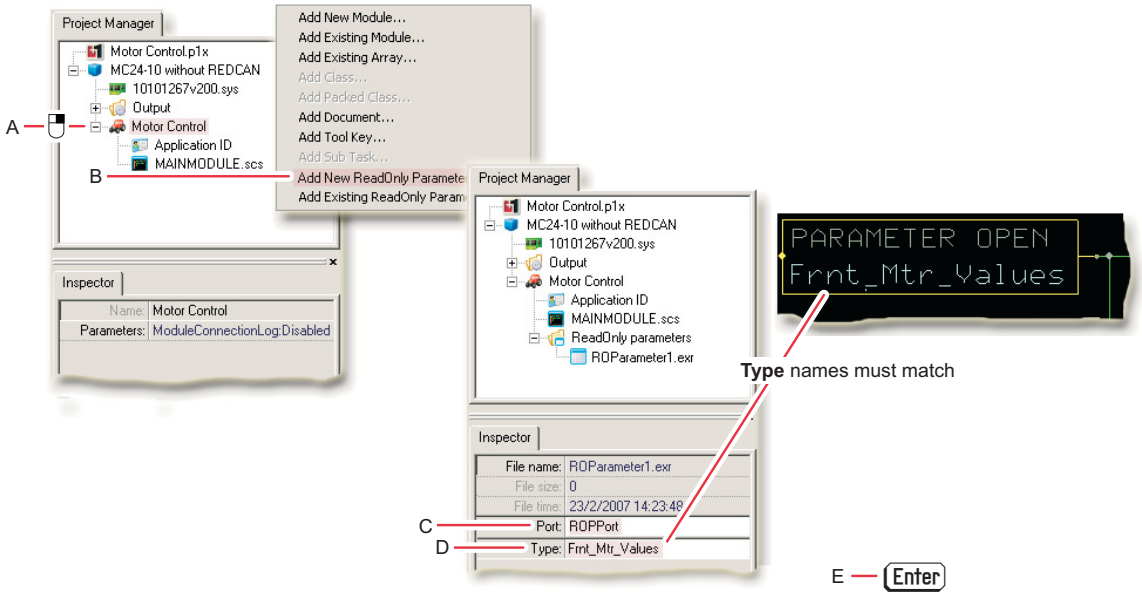
4. 読み取り専用パラメータ機能を有効化します。

   A. **Project Manager** タブで、***.SYS**（カーネル）ファイルをクリックします。

   B. **Inspector** タブで、**Parameters** プロパティのボタンをクリックして **Edit System Parameters** ウィンドウを表示させます。

   C. **Edit System Parameters** ウィンドウで、**Read-only Parameters** 機能を **ENABLE** に変更します。

   D. **Edit System Parameters** ウィンドウで、**OK** をクリックして **Parameters** の選択を適用します。



5. 読み取り専用パラメータ lhx ファイルを作成するために必要なプロパティを設定します。

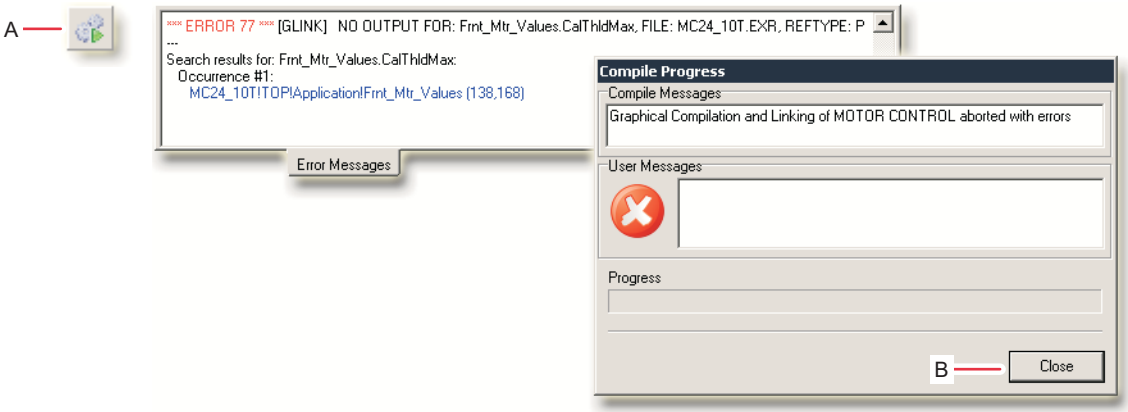
   A. **Project Manager** タブで、アプリケーション名（この例では **Motor Control**）を右クリックしポップアップメニューを表示させます。

   B. ポップアップメニューで **Add New Read-only Parameter** をクリックします。

   C. **Inspector** タブで、**Port** を選択します。

   D. **Inspector** タブで、**Type** の欄に **PARAMETER OPEN** コンポーネントで使用している **TYPE** 名（この例では **Frnt_Mtr_Values**）を入力します。

   E. Enterを押します。



6. CSV（カンマ区切り形式）のテンプレートファイルを生成します。

   A. PLUS+1 GUIDE アプリケーションツールバー上の Compile All ボタンをクリックします。

      コンパイルプロセスによって、

      – CSV テンプレートファイルが生成され、**Error Messages** タブに **ERROR 77** メッセージを表示して中断します。

      – アプリケーション内の読み取り専用パラメータファイルごとにエラーメッセージが表示されます。

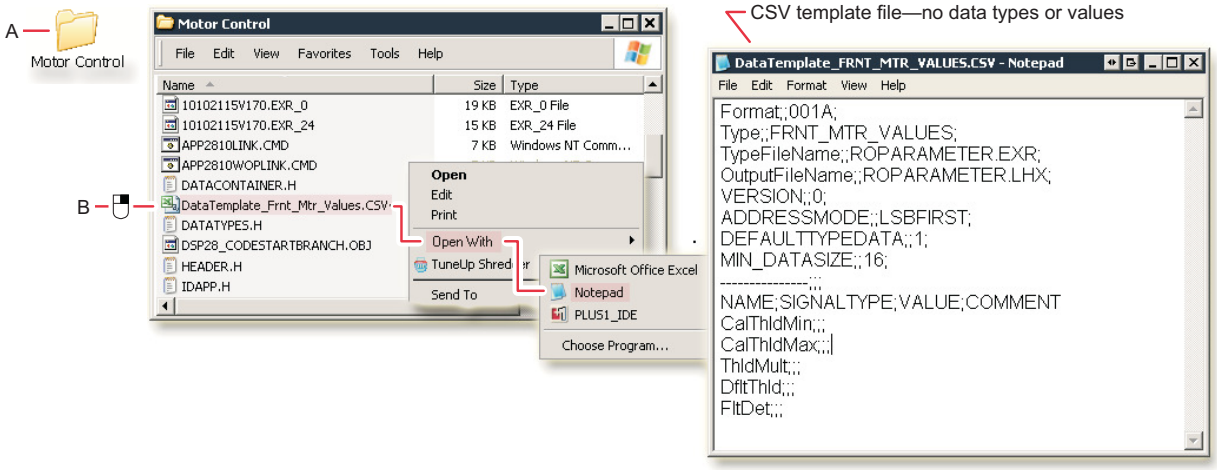
      CSV テンプレートファイルには各パラメータのフィールドが含まれますが、フィールドにはデータ型や値は含まれません。

      次のステップで、**ERROR 77** メッセージを解消するため、

      – CSV テンプレートファイルに値を入力し、名前をつけて保存して CSV ソースファイルを作成します。

      – PLUS+1 プロセスを走らせて CSV ソースファイル内の値をダウンロード可能読み取り専用パラメータ lhx ファイルとして出力します。

      B. **Close** をクリックして **Compile Progress** ウィンドウを閉じます。



7. CSV テンプレートファイルをブラウズして開きます。

   A. プロジェクトフォルダをブラウズし、CSV テンプレートファイルを探します。

      CSV テンプレートファイルには **DataTemplate_+TYPE**.csv という名前が付けられています。

      この例では、

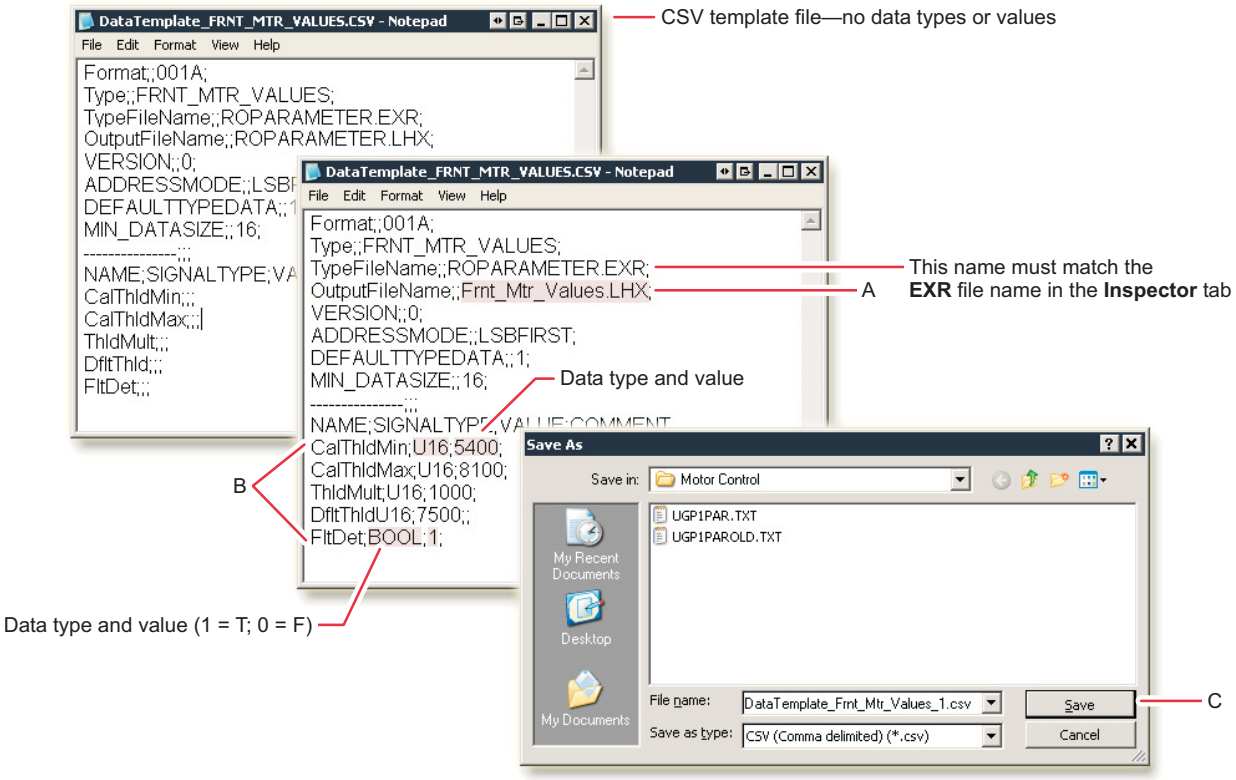
      – **PARAMETER OPEN** コンポーネントの **TYPE** 値は **Frnt_Mtr_Values** です。

      – CSV テンプレートファイルの名前は **DataTemplate_Frnt_Mtr_Values.csv** です。

   B. CSV テンプレートファイルを右クリックし、ポップアップメニューコマンドからテンプレートファイルを Notepad または Microsoft Excel で開きます。

      Microsoft Excel でも Notepad でも、区切り文字はセミコロン(;)です。

      CSVテンプレートファイルのフォーマットについては***453***ページの*異なる値を持つ読み取り専用パラメータファイルの作成*を参照してください。



8. CSV ソースファイルを作成します。

   A. 必要に応じて **OutputFileName** を変更します。

      OutputFileName はパラメータファイルの名前を定義します。

      この例ではパラメータファイルの名前を **ROPARAMETER.LHX** から **Frnt_Mtr_Values.lhx** に変更します。

   B. CSV テンプレートファイルでは、各パラメータのデータ型と値を入力します。

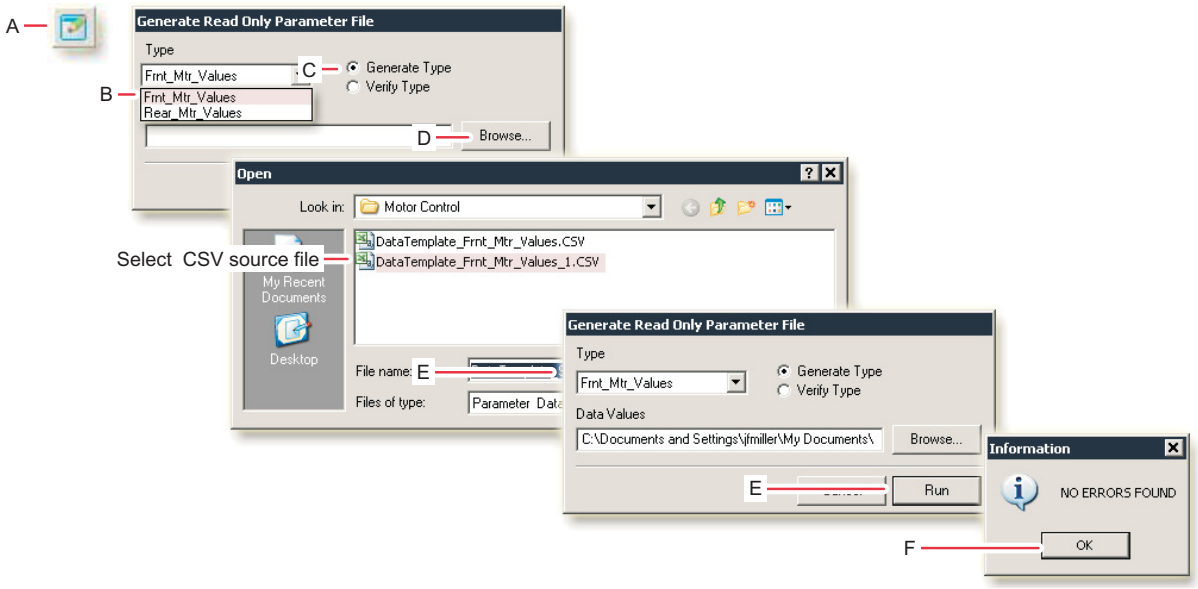
      データ型と値は必須、コメントはオプションです。

   C. 変更を加えた CSV ファイルを別の名前でセーブします。このファイルが CSV ソースファイルとなります。

      この例では、**DataTemplate_Frnt_Mtr_Values.csv** ファイルに追加されたパラメータ値が **DataTemplate_Frnt_Mtr_Values_1.csv** ファイルにセーブされます。

   CSV テンプレートを変更しないでください。すべての変更は CSV ソースファイルに加えてください。

   （複数の読み取り専用パラメータファイルを作成するときはステップ 7～8 を繰り返してください）



9. ダウンロード可能読み取り専用パラメータファイルを生成します。

   A. PLUS+1 GUIDE ツールバーで、Generate Parameter ボタンをクリックして **Generate Read-only Parameter File** ウィンドウを表示します。

   B. **Generate Read-only Parameter File** ウィンドウで、**Parameter Open** コンポーネントの **Type** 値にマッチした **Type** を選択します。

   C. **Generate Read-only Parameter File** ウィンドウで **Generate Type** をクリックします。

   D. **Generate Read-only Parameter File** ウィンドウで **Browse** をクリックして **Open** ウィンドウを表示します。

      Open ウィンドウを使用して CSV ソースファイルを選択します。

   E. **Generate Read-only Parameter File** ウィンドウで **Run** をクリックします。

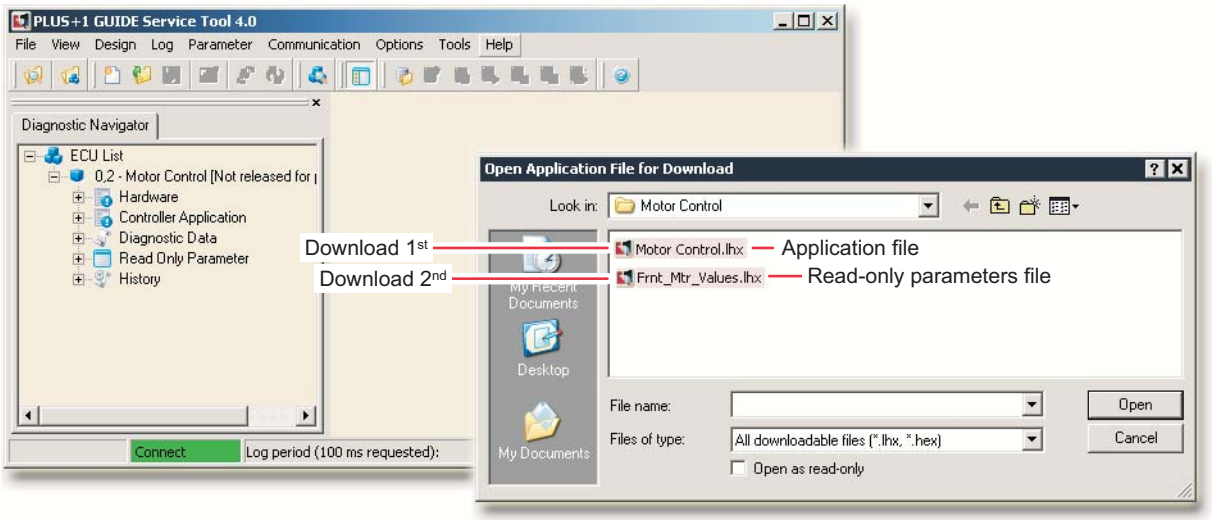
      PLUS+1 GUIDE プログラムが正常に CSV ソースファイルの値を読み取り専用ファイルに出力した場合、**Information** ウィンドウに **NO ERRORS FOUND** メッセージが表示されます。

   F. **Information** ウィンドウで **Ok** をクリックしてパラメータに関係する全てのウィンドウを閉じます。

10. アプリケーションをコンパイルします。

    ここで、プロジェクトには以下の二つの lhx ファイルが存在します。

    - アプリケーションファイル：この例では **Motor Control.lhx**
    - 読み取り専用パラメータファイル：この例では **Frnt_Mtr_Values.lhx**



11. サービスツールプログラムを開き、ファイルをダウンロードします。

- まずアプリケーションファイルをダウンロードします。

  アプリケーションファイルはコントローラのメモリロケーションを読み取り専用パラメータのためにそれらの **TYPE** 値、順序、データ型、名前の情報をリファレンスとして確保します。

- 次に読み取り専用パラメータファイルをダウンロードします。

  正しくファイルがダウンロードされるためには、ファイル内のリファレンスがコントローラメモリのリファレンスと一致する必要があります。

Lhx ファイルのダウンロードについての詳細は、***PLUS+1 Service Tool User Manual***.を参照してください。

異なる値を持つ読み取り専用パラメータファイルの作成

この手順では異なる値を持つ読み取り専用パラメータファイルを追加する方法を記述しています。既に読み取り専用パラメータファイルがあり、異なる値を持つファイルを追加したい場合はこの手順に従ってください。

この手順では以下のことを前提としています。

パラメータ値のみ変更が必要

パラメータ名は変更の必要なし

データ型は変更の必要なし

アプリケーション内のパラメータは追加あるいは削除されていない

アプリケーション内のパラメータの順序は変更されていない

以下の表は次のものを示しています。

既存の読み取り専用パラメータファイル内の古い値

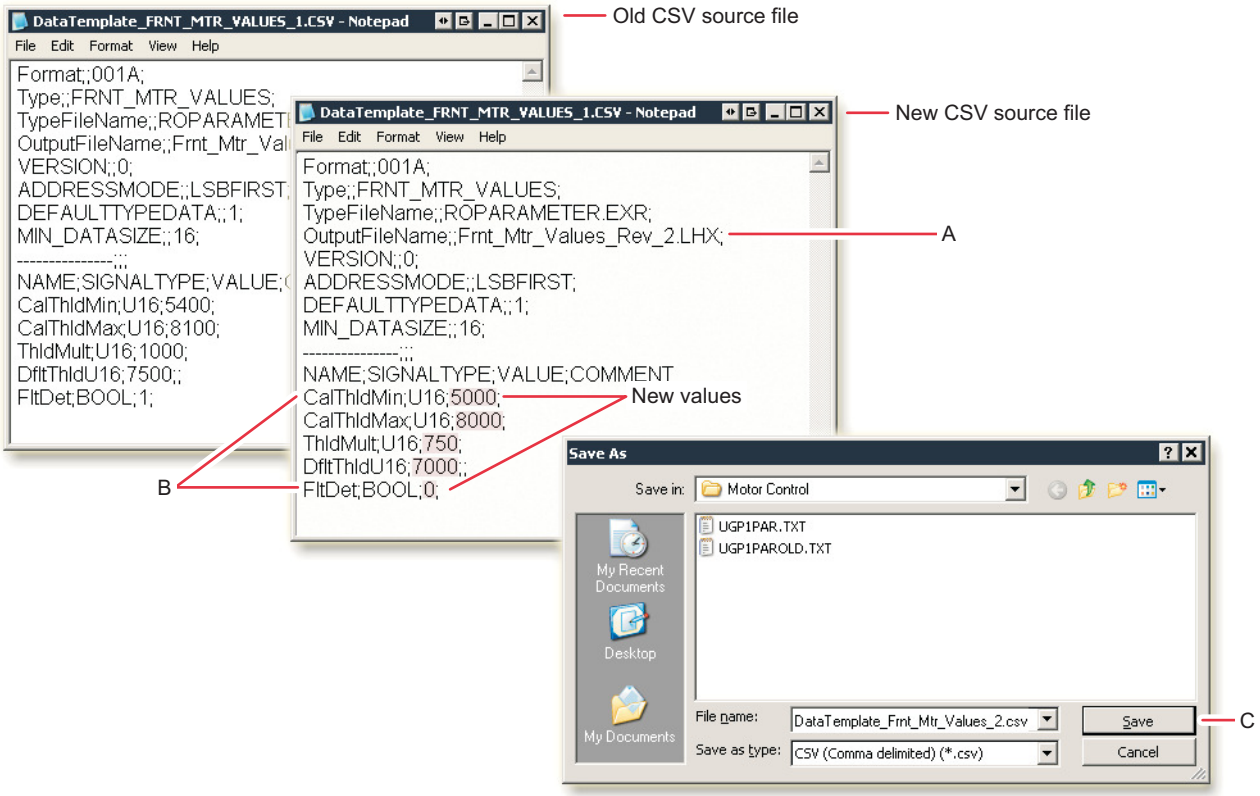
新しい読み取り専用パラメータファイル内の新しい値

読み取り専用パラメータ

名前	信号の型	古い値	新しい値
CalThldMin (Calibration Threshold Minimum)	U16	5400	5000
CalThldMax (Calibration Threshold Maximum)	U16	8100	8000
ThldMult (Threshold Multiplier)	U16	1000	750
DfltThld (Default Threshold)	U16	7500	7000
FltDet (Fault Detect)	BOOL	1 (1 =T; 0 =F)	0 (1 =T; 0 =F)

1. PLUS+1 GUIDE プログラムを開始します。

2. 追加の読み取り専用パラメータファイルを作成しようとする GUIDE プロジェクトを開きます。



3. CSV ソースファイル内のパラメータ値を変更します。

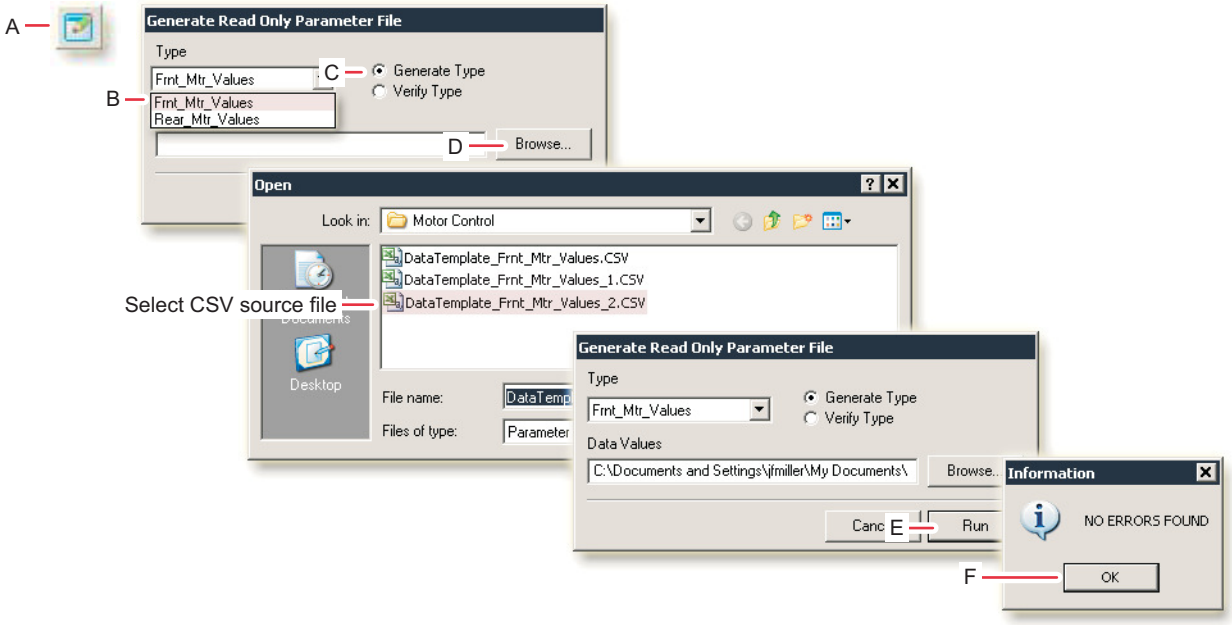
   A. 必要に応じて、**OutputFileName** を変更します。

      - **OutputFileName** は lhx パラメータファイルの名前を定義します。
      - この例では、パラメータファイルの名前を **Frnt_Mtr_Values.lhx** から **Frnt_Mtr_Values_Rev_2.lhx** に変更します。

   B. CSV ソースファイル内で各パラメータに新しい値を入力します。

   C. CSV ソースファイルを違うファイル名でセーブします。

      この例では、新しいパラメータセットは **DataTemplate_Frnt_Mtr_Values_2.csv** ファイルにセーブします。



4. ダウンロード可能読み取り専用パラメータファイルを生成します。

   A. ツールバーで、**Generate Parameter** ボタンをクリックして **Generate Read-only Parameter File** ウィンドウを表示します。

   B. **Generate Read-only Parameter File** ウィンドウで、**Type** を **PARAMETER OPEN** コンポーネント内で使われる **Type** 値に一致するように選択します。

   C. **Generate Read-only Parameter File** ウィンドウで **Generate Type** をクリックします。

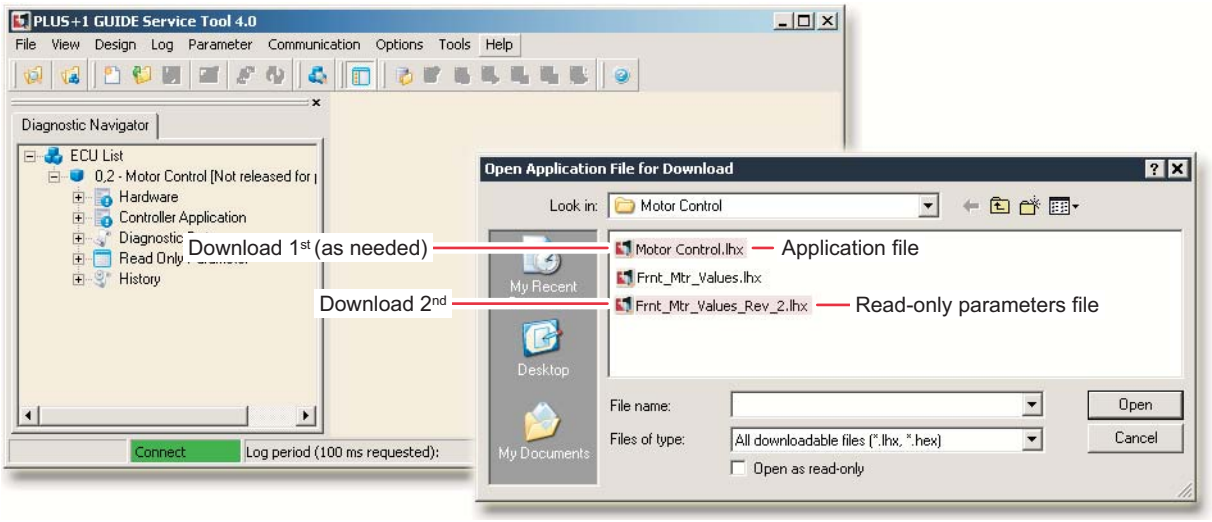
   D. **Generate Read-only Parameter File** ウィンドウで、**Browse** をクリックして **Open** ウィンドウを表示します。

      Open ウィンドウを使用して読み取り専用パラメータファイルで使用される **Data Values** のための CSV ソースファイルを選択します。

   E. **Generate Read-only Parameter File** ウィンドウで **Run** をクリックします。

      PLUS+1 GUIDE プログラムが CSV ソースファイル内の値を正常に読み取り専用パラメータファイルに出力した場合、**Information** ウィンドウに **NO ERRORS FOUND** メッセージが表示されます。

   F. **Information** ウィンドウで **OK** をクリックして、全てのパラメータに関連したウィンドウを閉じます。



5. サービスツールプログラムを開き、ファイルをダウンロードします。
   - 最初にアプリケーションファイルをダウンロードします（まだダウンロードしていない場合）。
   - 次に読み取り専用パラメータファイルをダウンロードします。

パラメータを変更したアプリケーションのための新規読み取り専用パラメータファイルの作成

この手順では、読み取り専用パラメータファイルを持つアプリケーションにおいてパラメータを変更した後で新たに読み取り専用パラメータファイルを作成する方法を記述しています。パラメータを変更（追加、削除、名前の変更、順序の変更）し、新しく読み取り専用パラメータファイルを作成したい場合にはこの手順に従ってください。

読み取り専用パラメータファイルを持つ既存のアプリケーションで、以下のことを行うときはこの手順に従ってください。

- パラメータの追加あるいは削除
- パラメータ名の変更
- パラメータのデータ型の変更

以下の表は次のことを示しています。

オリジナルのアプリケーションのパラメータと読み取り専用パラメータファイルの値

修正したアプリケーションのパラメータと読み取り専用パラメータファイルの値

BkUpAlarm (Backup Alarm)は新規パラメータです。

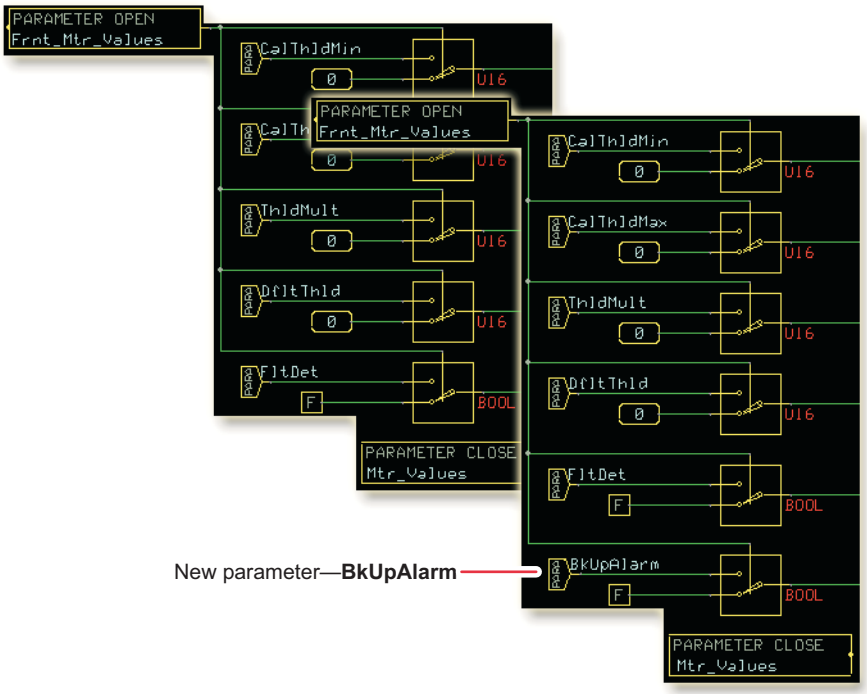
オリジナルの読み取り専用パラメータ

名前	信号の型	値
CalThldMin (Calibration Threshold Minimum)	U16	5400
CalThldMax (Calibration Threshold Maximum)	U16	8100
ThldMult (Threshold Multiplier)	U16	1000
DfltThld (Default Threshold)	U16	7500
FltDet (Fault Detect)	BOOL	1 (1 =T; 0 =F)

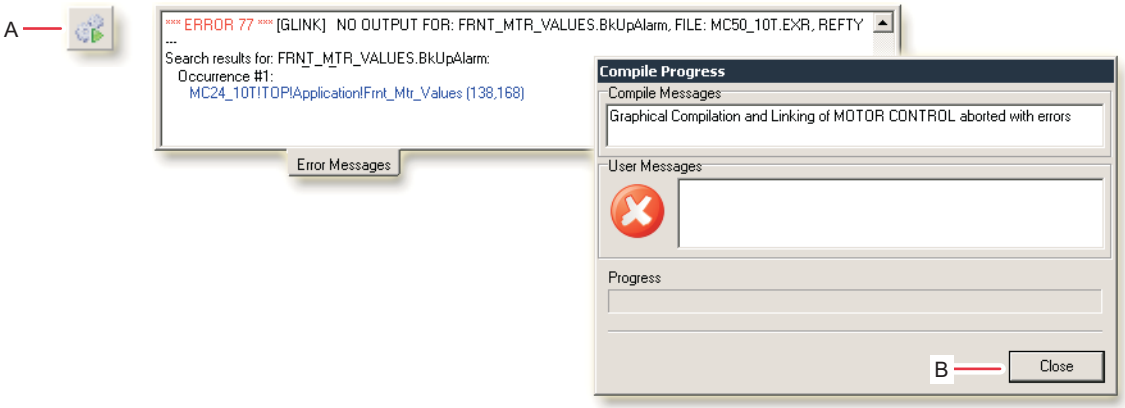
修正した読み取り専用パラメータ

名前	信号の型	値
CalThldMin (Calibration Threshold Minimum)	U16	5400
CalThldMax (Calibration Threshold Maximum)	U16	8100
ThldMult (Threshold Multiplier)	U16	1000
DfltThld (Default Threshold)	U16	7500
FltDet (Fault Detect)	BOOL	1 (1 =T; 0 =F)
BkUpAlarm (Backup Alarm)	BOOL	1 (1 =T; 0 =F)

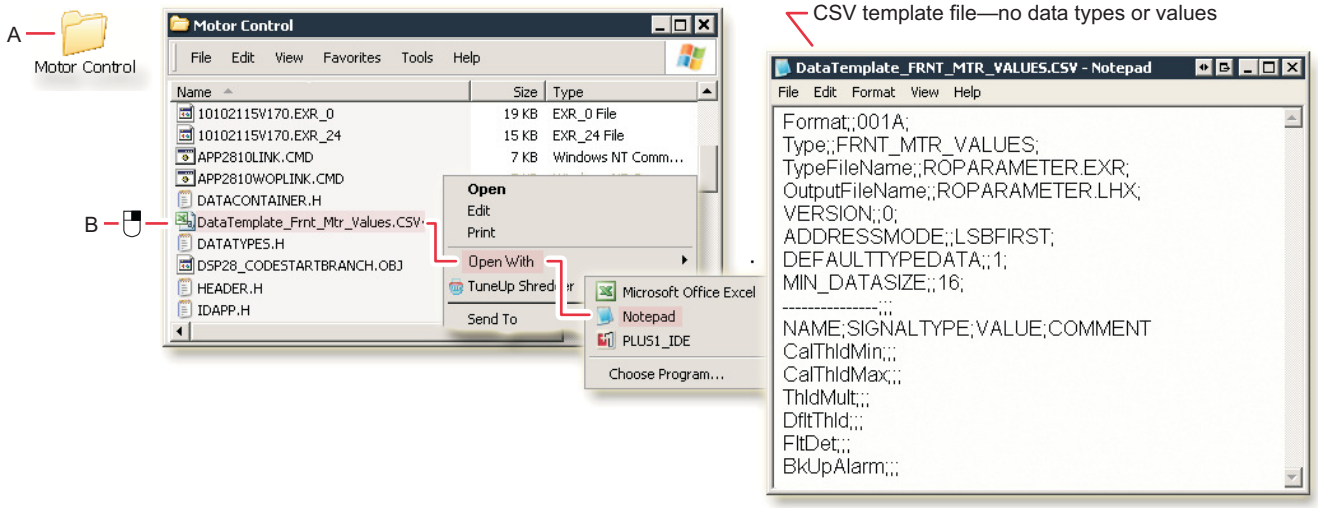
1. PLUS+1 GUIDE プログラムを開始します。
2. 読み取り専用パラメータファイルを作成したい GUIDE プロジェクトを開きます。



3. 追加パラメータを読み取りパラメータのロジックに追加します。



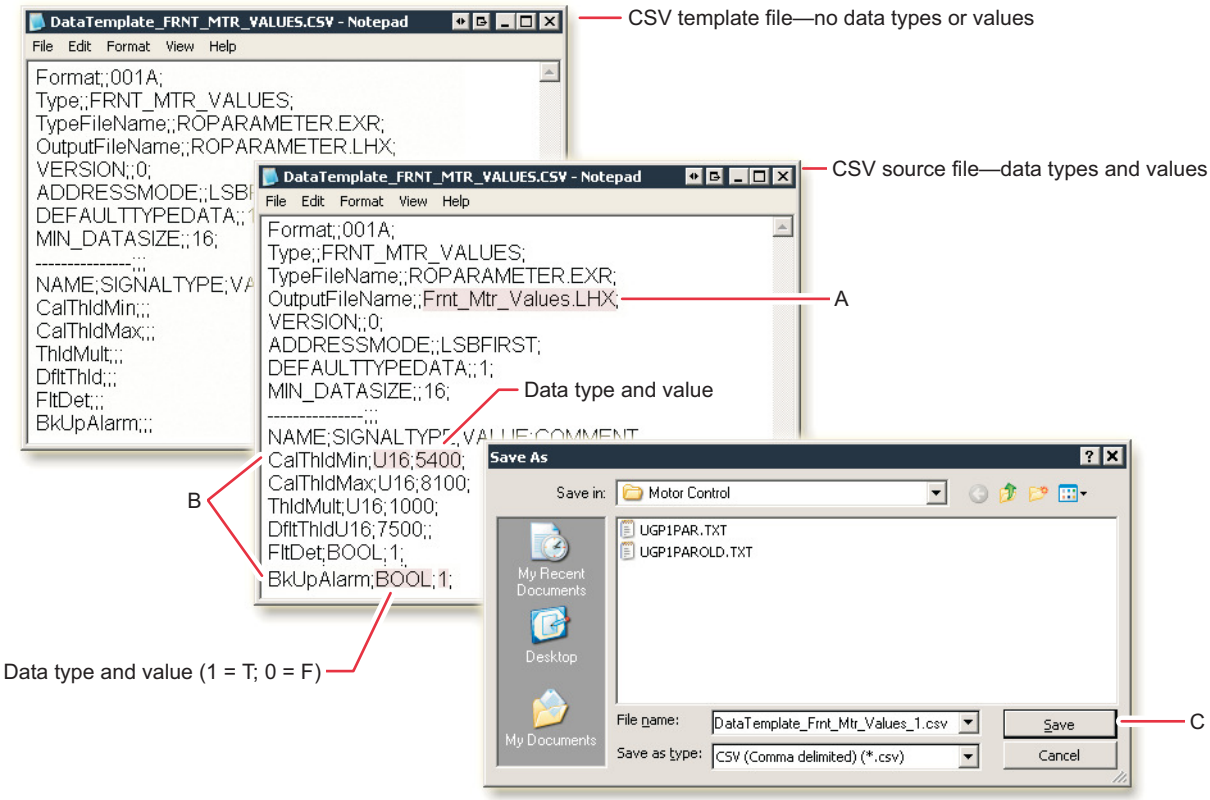
4. CSV（カンマ区切り）テンプレートファイルを生成します。
   A. PLUS+1 GUIDE アプリケーションツールバーで、Compile All ボタンをクリックします。
   B. **Close** をクリックして **Compile Progress** ウィンドウを閉じます。



5. CSV テンプレートファイルをブラウズして開きます。

   A. プロジェクトフォルダをブラウズして CSV テンプレートファイルの場所を探します。

      B. CSV テンプレートファイルを右クリックし、ポップアップメニューコマンドを使用してファイルをメモ帳または Microsoft Excel で開きます。



6. CSV ソースファイルを作成します。

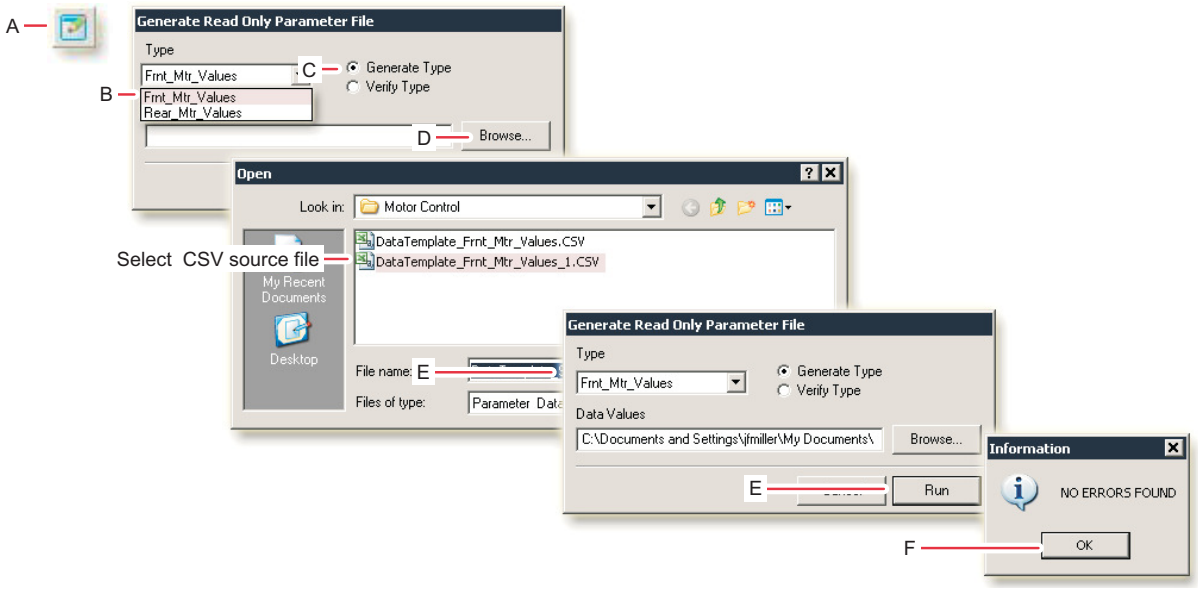
   A. 必要に応じて **OutputFileName** を変更します。

      OutputFileName はパラメータファイルの名前を定義します。

   B. CSV テンプレートファイルで、データ型と各パラメータの値を入力します。

   C. 変更した CSV テンプレートファイルを別のファイル名で保存します。このファイルが CSV ソースファイルとなります。

   CSV テンプレートファイルを変更しないでください。変更は全て CSV ソースファイル内で行ってください。



7. ダウンロード可能読み取り専用パラメータファイルを生成します。

   A. PLUS+1 GUIDE ツールバーで Generate Parameter ボタンをクリックして **Generate Read-only Parameter File** ウィンドウを表示します。

   B. **Generate Read-only Parameter File** ウィンドウで **Parameter Open** コンポーネントの **Type** 値と一致する **Type** を選択します。

   C. **Generate Read-only Parameter File** ウィンドウで **Generate Type** をクリックします。

   D. **Generate Read-only Parameter File** ウィンドウで **Browse** をクリックして **Open** ウィンドウを表示します。

      Open ウィンドウを使用して CSV ソースファイルを選択します。

   E. **Generate Read-only Parameter File** ウィンドウで **Run** をクリックします。

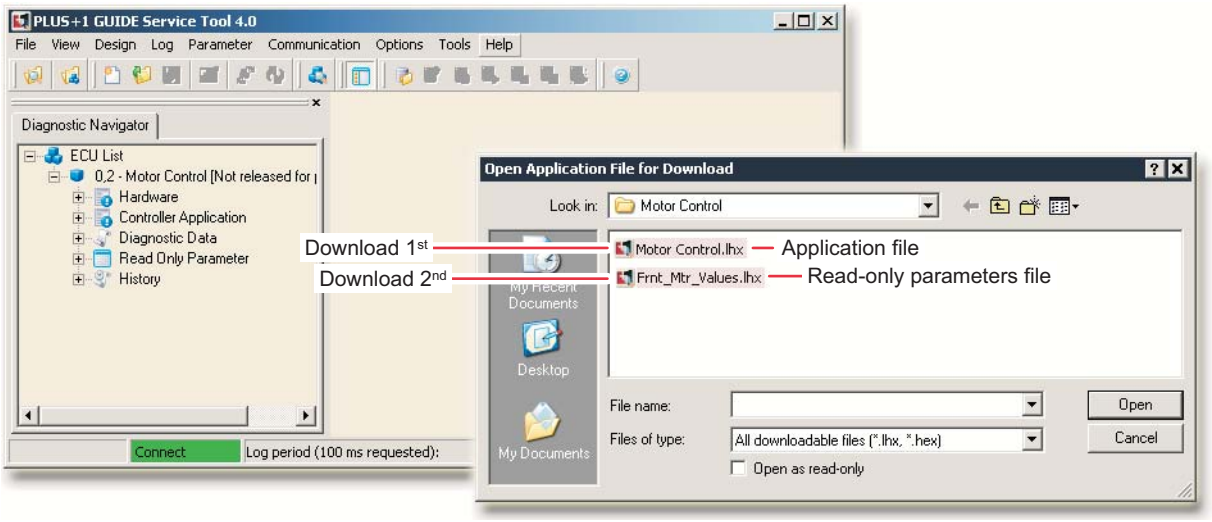
      PLUS+1 GUIDE プログラムが正常に CSV ソースファイル内の値を読み取り専用パラメータファイルに出力した場合、**Information** ウィンドウに **NO ERRORS FOUND** メッセージが表示されます。

   F. **Information** ウィンドウで **OK** をクリックして全てのパラメータに関係するウィンドウを閉じます。

8. アプリケーションをコンパイルします。

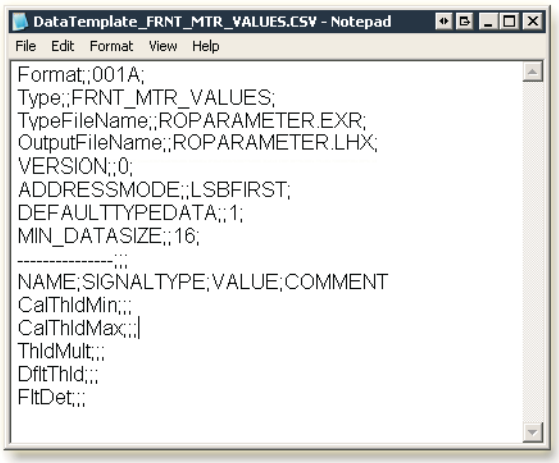
   現在、プロジェクトには 2 つのダウンロード可能 lhx ファイルがあります。

   - アプリケーションファイル: この例では **Motor Control.lhx**
   - 読み取り専用パラメータファイル: この例では **Frnt_Mtr_Values.lhx**



9. サービスツールプログラムを開き、ファイルをダウンロードします。
   - 最初にアプリケーションファイルをダウンロードします。
   - 次に読み取り専用パラメータファイルをダウンロードします。

CSVテンプレートファイルのフォーマットについて



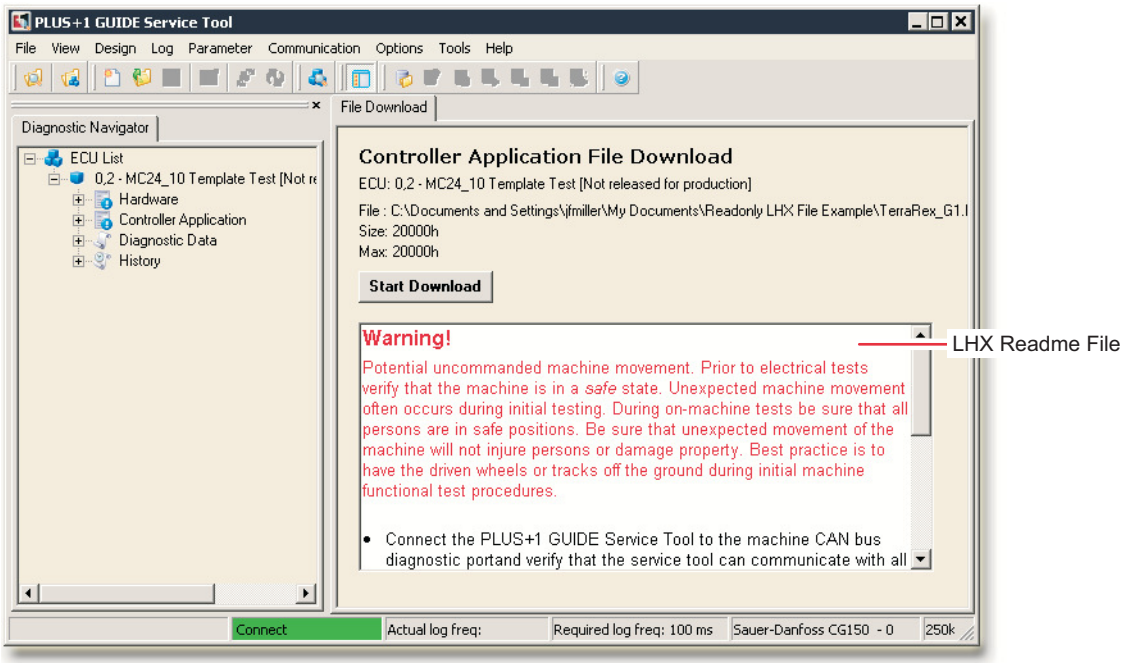
CSV テンプレートファイルには各読み取り専用パラメータに対応するフィールドはありますが、値は含まれていません。ユーザはこれらのフィールドに値を書き込み、新しく名前をつけて保存することで CSV ソースファイルを作成します。CSV ソースファイルはダウンロード可能読み取り専用パラメータ lhx ファイルの元ファイルとなります。

CSV テンプレートファイルフォーマット

行	ラベル	フィールドの内容
1	**Format**	読み取り専用パラメータファイルのフォーマットを指定します。ARRAY 型のサポートのためにこの型を用います。
2	**Type***	**PARAMETER OPEN** コンポーネントの **TYPE** 値を使って読み取り専用パラメータセットを指定します。
3	**TypeFileName***	読み取り専用パラメータファイルを生成するために使う*EXR ファイルを指定します。
4	**OutputFileName**	**Generate Read-only Parameter File** ウィンドウで作成するダウンロード可能読み取り専用パラメータ lhx ファイルの名前を定義します。必要に応じて新しい出力ファイル名を入力してください（この例では **ROPARAMETER.LHX** をもっと分かりやすい名前に変更できます）。
5	**VERSION**	PLUS+1 サービスツールプログラム上で表示されるバージョンフィールドを定義します。必要に応じてバージョンを入力してください。
6	**ADDRESSMODE***	ハードウェアによって定義されるアドレスモード
7	**DEFAULTTYPEDATA***	ハードウェアによって定義されるデフォルトデータ型
8	**MIN_DATASIZE***	ハードウェアによって定義される最小データサイズの値
9	**– – – – –***	ファイルヘッダとデータの区切り
10	**NAME*; SIGNALTYPE; VALUE; COMMENT**	以降の行のデータヘッダ
11–		アプリケーション内で設定されたパラメータから生成された読み取り専用パラメータ。各読み取り専用パラメータに対し、**SIGNALTYPE**（データ型）と **VALUE** を入力します。必要に応じてオプションの **COMMENT** を入力します。コメントは読み取り専用パラメータ lhx ファイルには含まれません。常にセミコロン(;)でパラメータを区切ってください。 配列に関しては、以下のいずれかのフォーマットを使ってください。For arrays, use one of the following formats: – ArrVariable1;ARRAY[3]U8;(1,4,9);My first comment – ArrVariable2;ARRAY[3]U8;1;4;9;My first comment（CSV ファイルを Excel 形式のファイルから生成する場合）

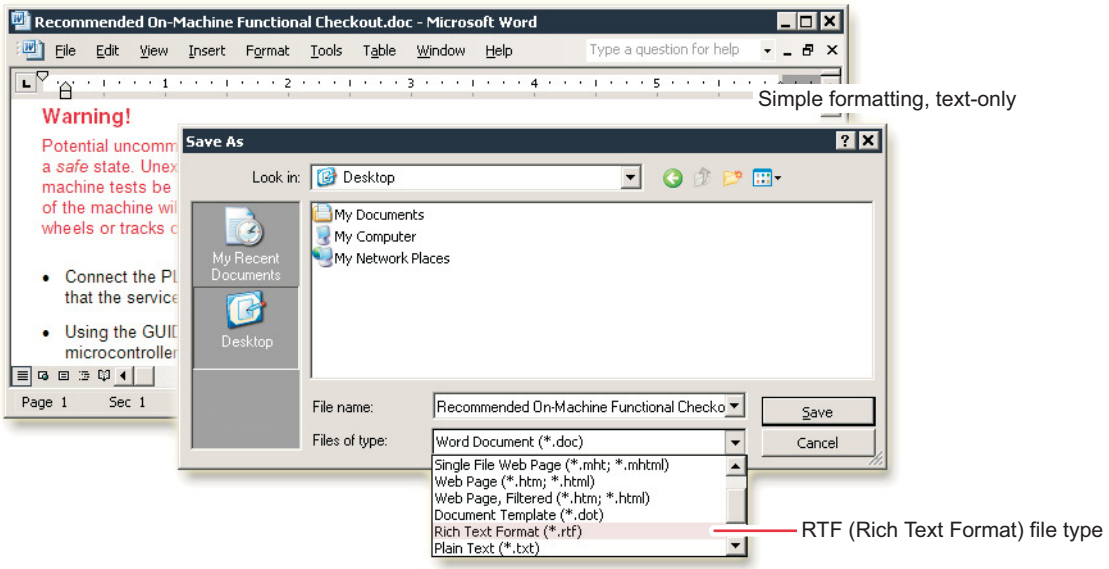
*変更しないでください。

Readme LHX ファイルの追加



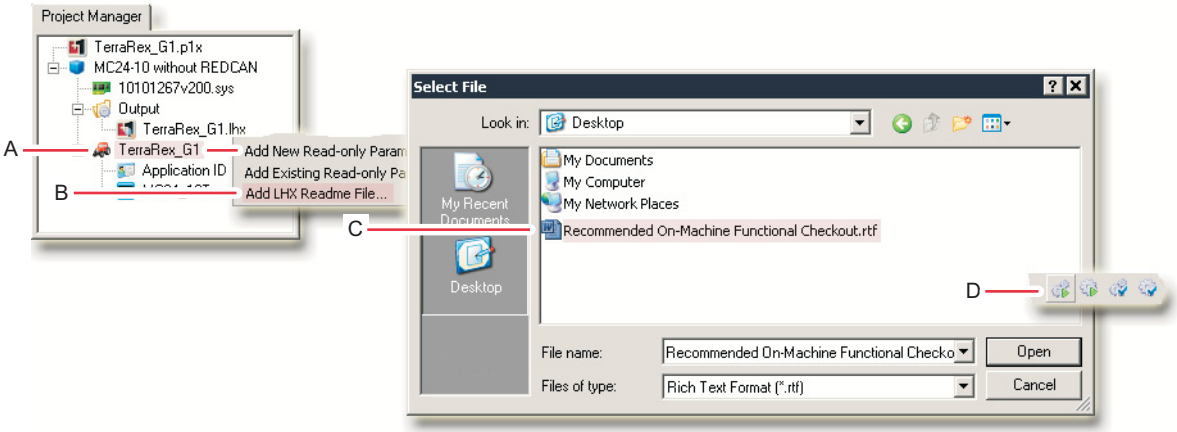
PLUS+1 Service Tool は LHX ファイルをダウンロードする直前に readme LHX ファイルを表示させることができます。

LHX ダウンロードファイルに readme ファイルを添付するには以下のステップを実行してください。



1. readme 文書を作成します。

   Microsoft Word または Wordpad を用いてテキストオンリー形式の readme 文書を作成してください。その文書を RTF（リッチテキスト形式）ファイルとして保存します。



2. readme ファイルを LHX ファイルに添付します。

   A. **Project Manager** タブで、Application Name を右クリックしてポップアップメニューを表示させます。

   B. ポップアップメニューで、**Add LHX Readme File** をクリックして **Select File** ウィンドウを表示させます。

   C. **Select File** ウィンドウで RTF フォーマットの readme ファイルを選択します。

   D. LHX ファイルをコンパイルします。

パーツ番号やシリアル番号によるダウンロードの制限

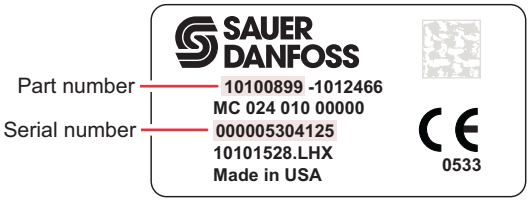
LHX ロック機能により、以下のいずれかのダウンロード制限を掛けることができます。

- パーツ番号による制限

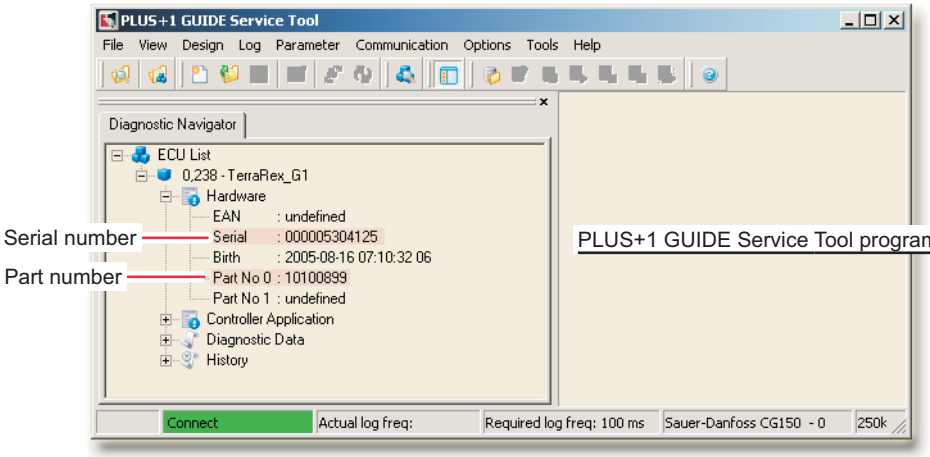
  パーツ番号が一致した PLUS+1 機器にのみ LHX アプリケーションファイルをダウンロード可能

- シリアル番号による制限.

  シリアル番号が一致した PLUS+1 機器（1 台）にのみ LHX アプリケーションファイルをダウンロード可能

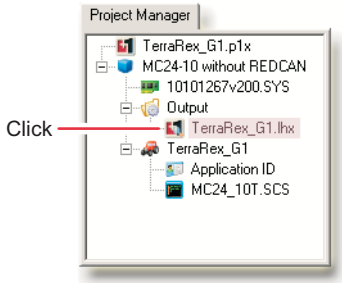


Part and serial number on a typical hardware label

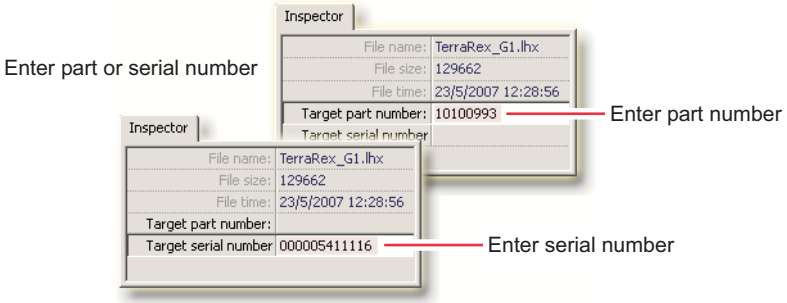


PLUS+1 GUIDE Service Tool program displays serial and part number

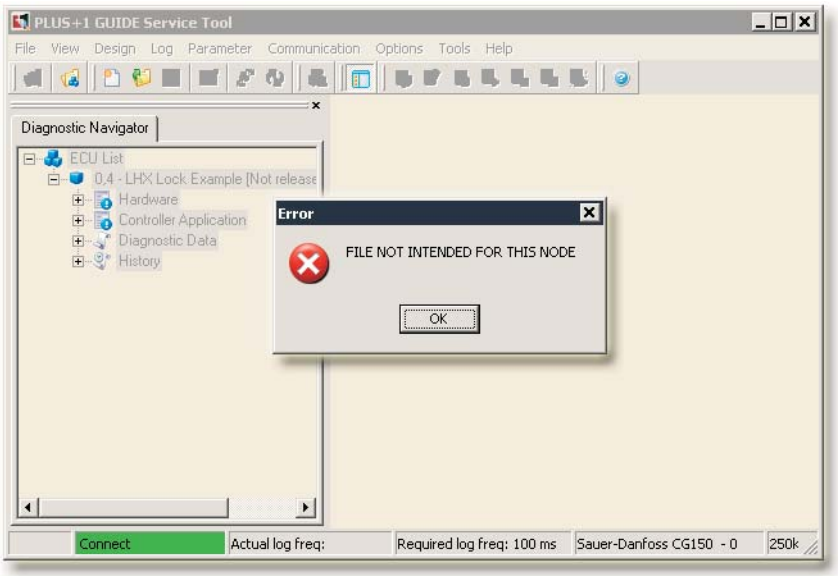
パーツ番号またはシリアル番号によって LHX アプリケーションファイルのダウンロードに制限を掛けるには、以後の手順を実行してください。



1. **Project Manager** タブで lhx アプリケーションファイルをクリックします。



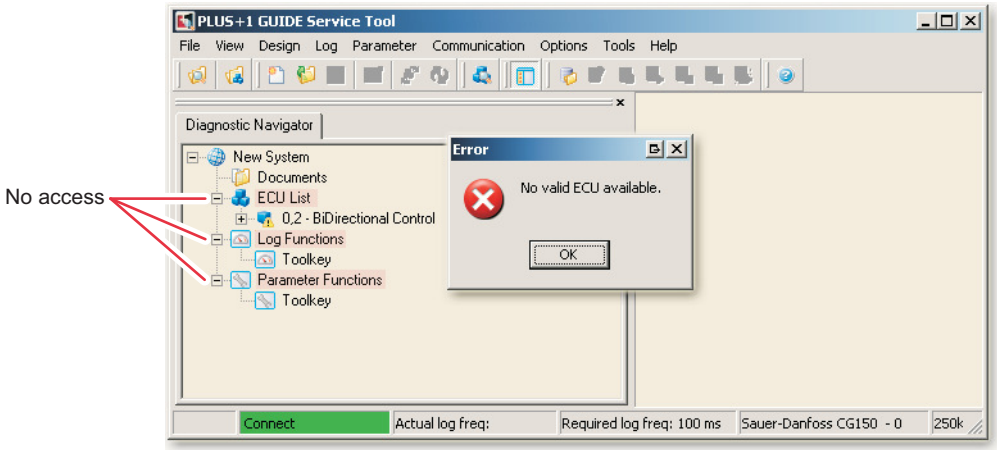
2. **Inspector**タブで、以下のいずれかを入力します。
   - LHX アプリケーションファイルをダウンロード可能にしたい PLUS+1 機器の **Target part number** 、または、
   - LHX アプリケーションファイルをダウンロード可能にしたい PLUS+1 機器の **Target serial number** 。
3. アプリケーションをコンパイルします。.



- 以下のどちらかが一致した場合、通常通り LHX アプリケーションファイルのダウンロードプロセスが進行します。
  - **Target part number** と機器のパーツ番号、または、
  - **Target serial number** と機器のシリアル番号
- 部品番号もシリアル番号も一致しない場合、LHX アプリケーションファイルのダウンロードは失敗し、**Error** メッセージが表示されます。

アプリケーション内の変数へのサービスツールによるアクセスの制限

Toolkey 機能により PLUS+1 サービスツールプログラムがアプリケーション内の変数にアクセスすることを制限することができます。



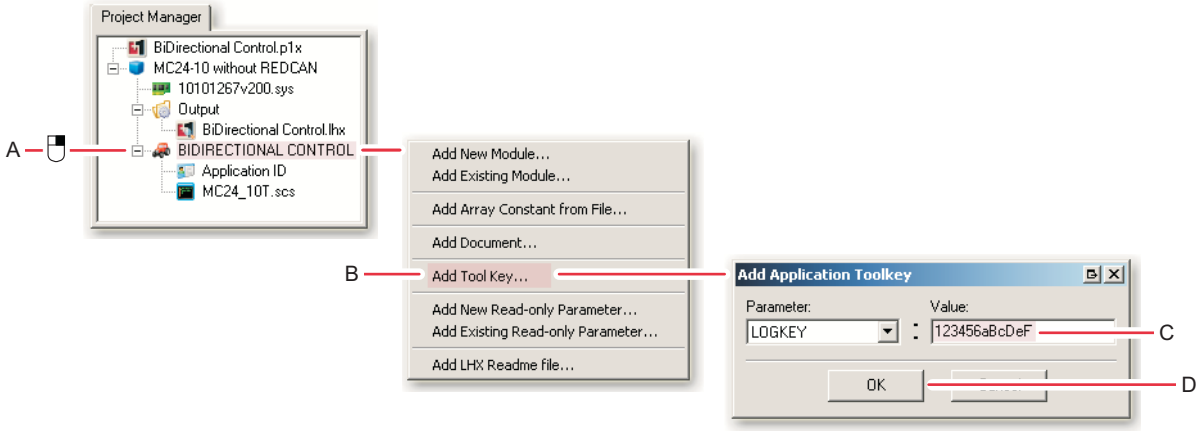
Toolkey 機能はユーザ定義の「Toolkey」(パスワード)を LHX ダウンロードファイル内に埋め込みます。

サービスツールプログラムは、

- Toolkey で保護されたアプリケーションをダウンロードするのに Toolkey は不要です。
- Toolkey で保護されたアプリケーションの値を読み取ったり書き込んだりするのに Toolkey が必要です。

  正しい Toolkey がない場合、サービスツールプログラムは ECU リスト、ログ機能、パラメータ機能へのアクセスができません。

LHX ダウンロードファイルに Toolkey を埋め込むには以後の手順を行います。



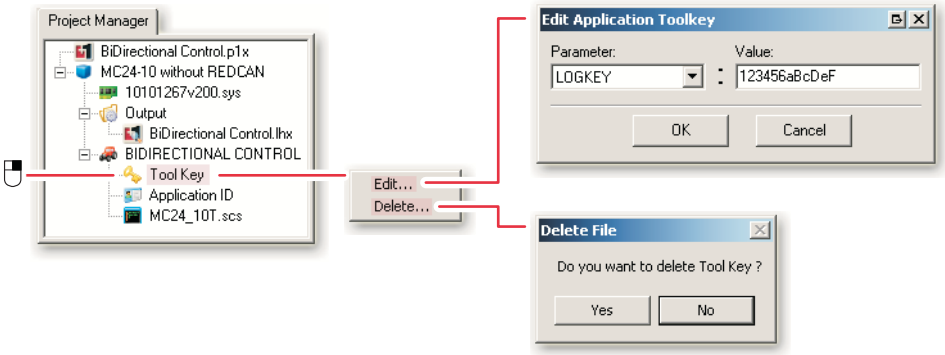
1. PLUS+1 GUIDE プロジェクトに Toolkey を追加します。

   A. **Project Manager** タブで、Application Name を右クリックしてポップアップメニューを表示させます。

   B. ポップアップメニューで、**Add Tool Key** をクリックして **Add Application Toolkey** ウィンドウを表示させます。

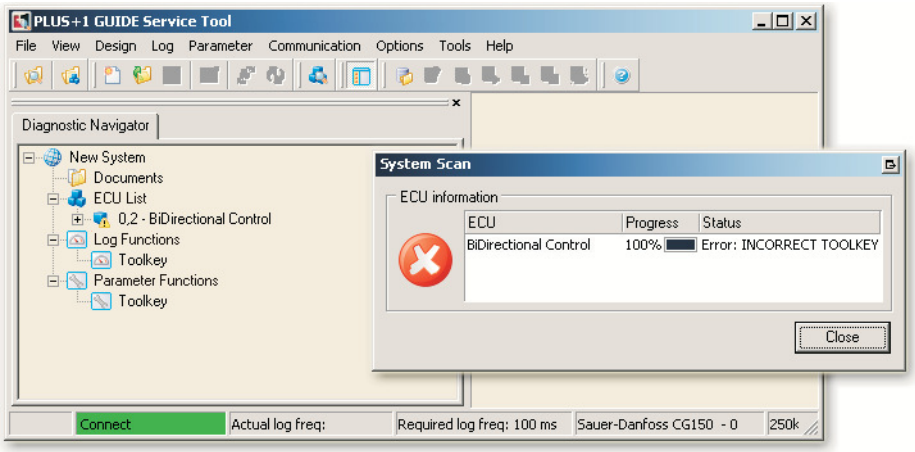
   C. **Add Application Toolkey** ウィンドウで **Value** に Toolkey（パスワード）を入力します。

   D. **Add Application Toolkey** ウィンドウで **OK** をクリックします。

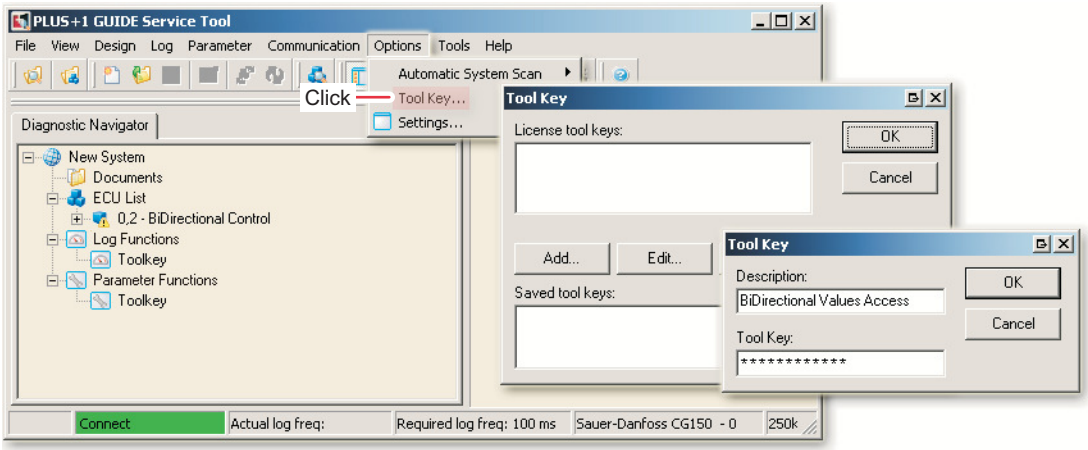


   Page Navigator タブに **Tool Key** エントリが表示されます。必要に応じてこのエントリを右クリックして Toolkey を編集したり削除したりすることができます。

2. プロジェクトをコンパイルして Toolkey の値を LHX ダウンロードファイルに埋め込みます。



サービスツールプログラムが Toolkey で保護されたアプリケーションをダウンロードした後、**System Scan** ウィンドウが **INCORRECT TOOLKEY** メッセージを表示します。



Option メニューの **Tool Key** コマンドを使って、アプリケーションの値へのアクセスに必要な Toolkey を入力してください。

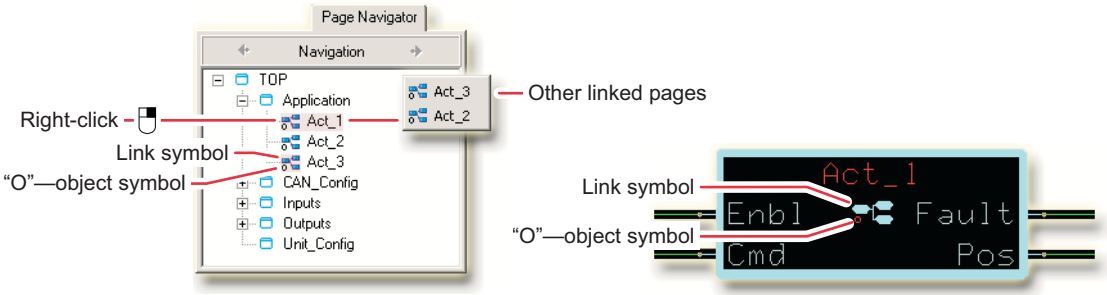
リンクページの作成

リンク機能により、以下の操作によって複製されたページが自動的にリンクされます。

Copy/Repeat コマンド

- **Copy to Clipboard** あるいは **Paste from Clipboard** コマンド
- **Import** コマンド（2 つ以上の同一のページが同じアプリケーションにインポートされたとき）

リンクページの見分け方



Page Navigator タブで、

- リンクページの横に小さくリンクのシンボルが現れます

  リンクページを右クリックすると、そのページの他のリンクがポップアップ表示されます

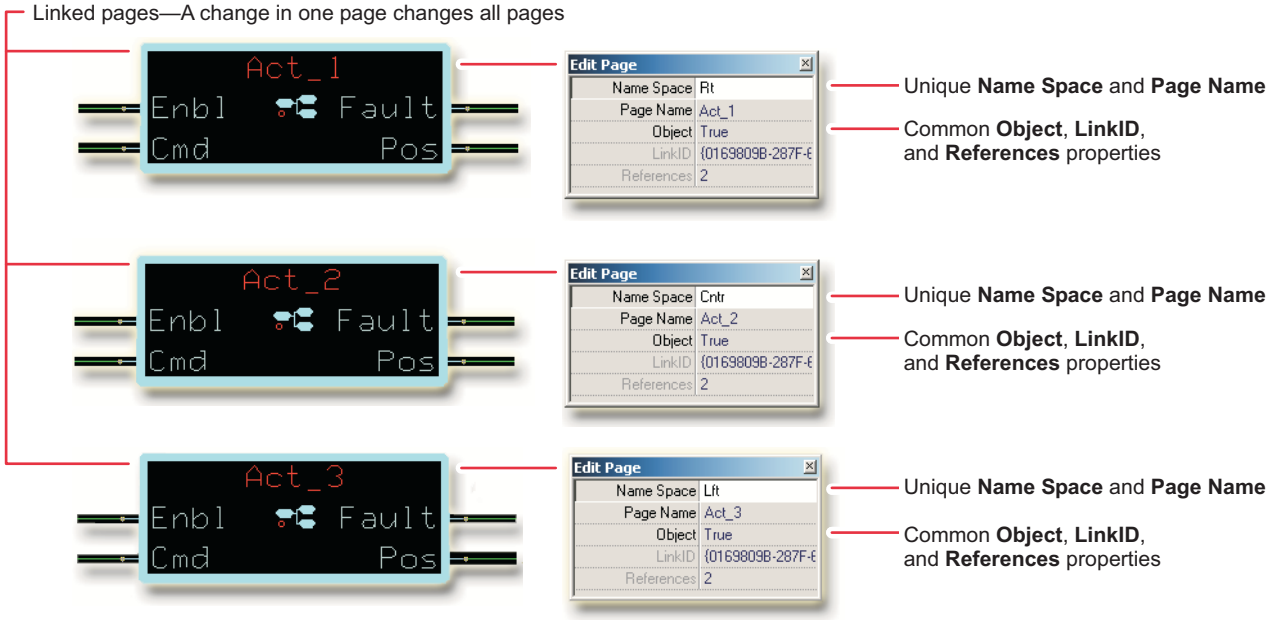
- リンクページの **Object** プロパティが **True** の場合、"O"がリンクシンボルの下に表示されます

ページのトップビューで、

- リンクされた各ページ内に小さなリンクシンボルが現れます
- リンクページの **Object** プロパティが **True** の場合、"O"がリンクシンボルの横に表示されます

  オブジェクトページについての詳細な情報は、***477***ページの*リンクページのオブジェクトページへの変換* を参照してください。

リンクページについて

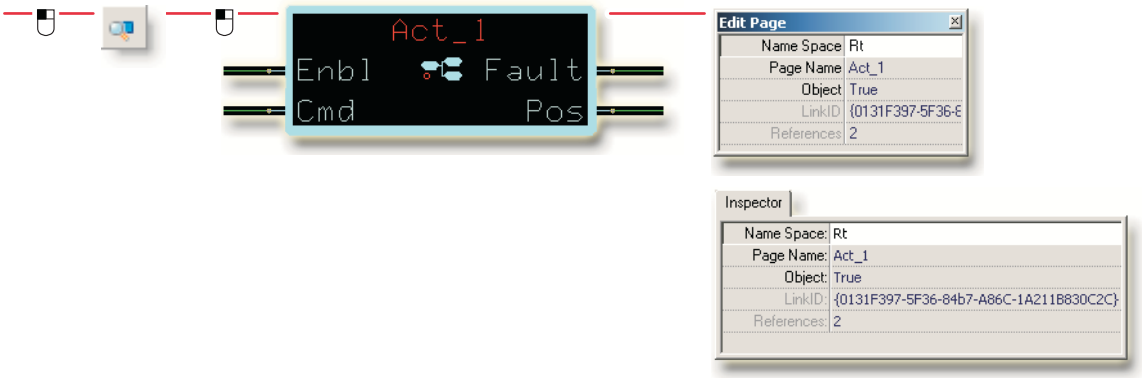


リンク機能により、一つのページ内の変更は以下のように反映されます

- リンクページの一つで変更された内容は、他のリンクページにも反映されます。
- 一つのリンクページのトップビューページの要素（ラベル、ポート名、ポートの型）を変更すると、他のリンクページのトップビューの要素も変更されます。
- 一つのリンクページの **Object** プロパティを変更すると、他の全てのリンクページの **Object** プロパティも変更されます。

リンク機能は、リンクページの **Page Name** や **Name Space** プロパティに影響を与えません。これらのプロパティはページごとに独立しています。

リンクページのプロパティ

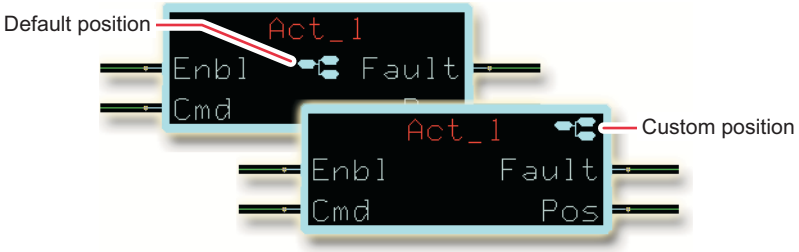


リンクページを Query してポップアップ **Edit Page** ウィンドウまたは **Inspector** タブ内でプロパティを確認することができます。

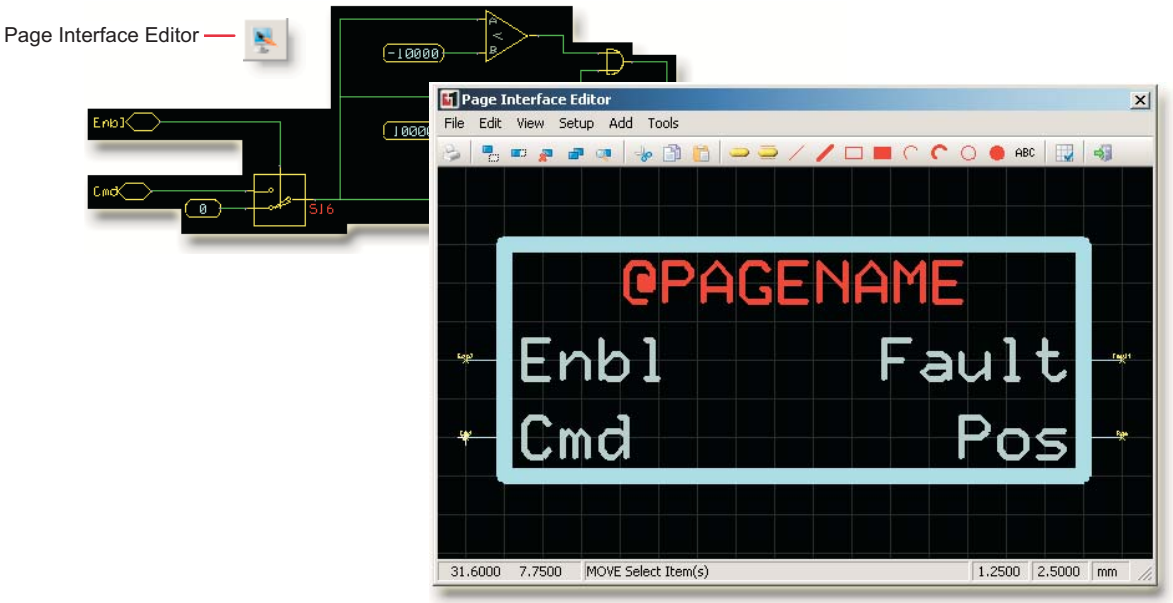
リンクページのプロパティ

プロパティ	説明
Name Space	メモリやチェックポイント名に個別のプレフィックスを適用し、名前の割り当てに起因するコンパイラエラーを回避します。 リンクページにチェックポイントやメモリコンポーネントを含む場合は、常に各ページに **Name Space** を割り当ててください。
Page Name	ページの名前。リンクされた各ページにはそれぞれ個別の名前を割り当てることができます。
Object	オブジェクトページは通常、フラッシュメモリ消費が少なく、RAM メモリ消費が少し多いコンパイルコードを生成します。 T = コンパイラは、リンクページのコンパイルコードを複製せずに追加インスタンスを扱います F = コンパイラは、リンクページのコンパイルコードを複製して追加インスタンスを扱います
LinkID	PLUS+1 GUIDE プログラム内部でリンクページのセットをトラックするために使用する共通 ID です。 **Inspector** タブにフル **LinkID** 番号が表示されます。
References	Query したページにリンクされているページの数です。 （例えば、**References** 値が **2** の場合は全部で 3 つのリンクページが存在することになります。）

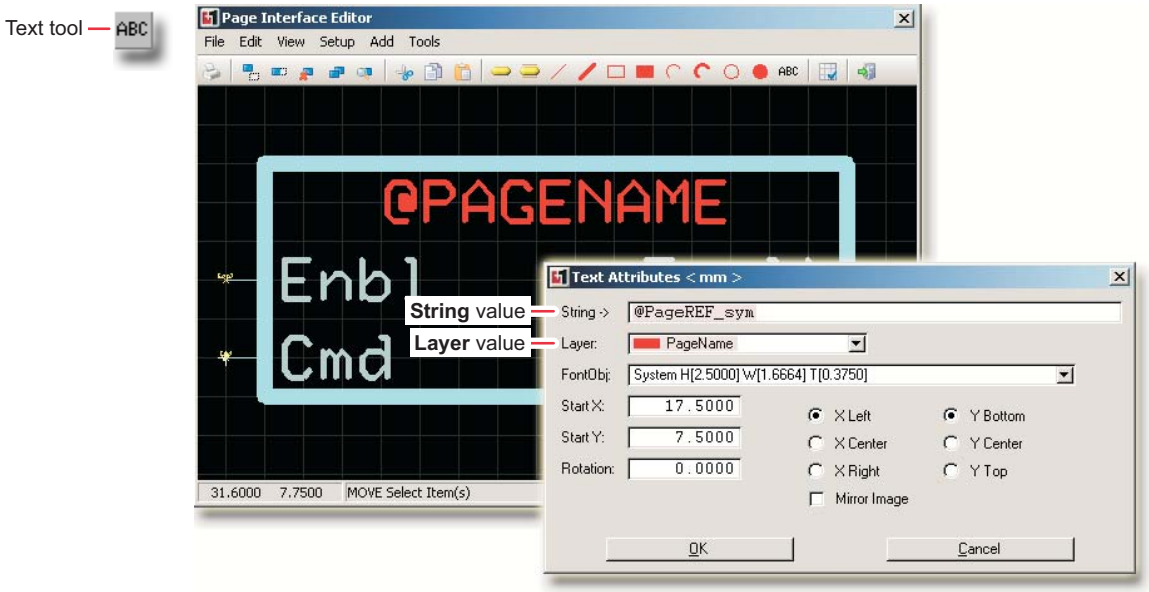
リンクシンボルの位置の変更



リンクシンボルはデフォルトの位置（ページ中央）から所望の位置に移動させることができます。



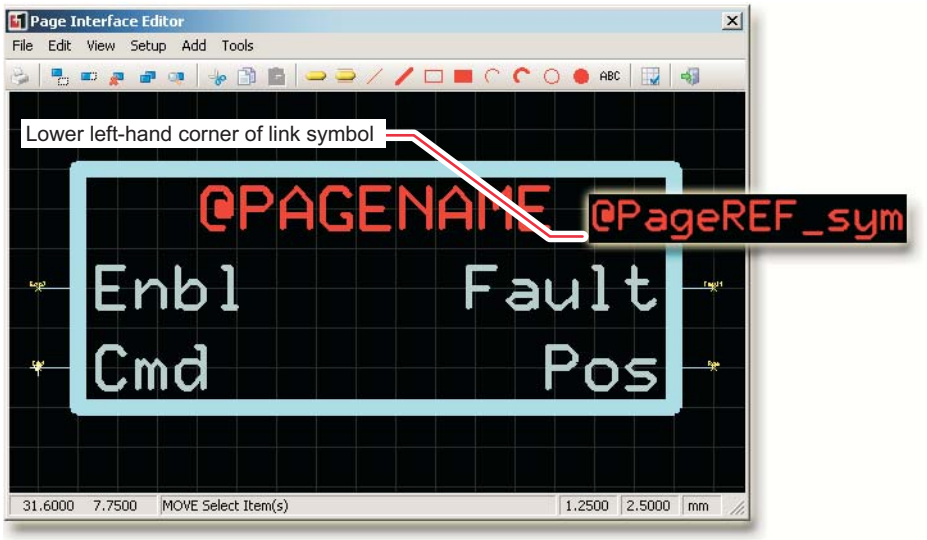
1. リンクシンボルの位置を移動したいページのトップビューを編集するため、**Page Interface Editor** ウィンドウに入ります。



2. Text ツールをクリックして **Text Attributes** ウィンドウを表示します。

   Text Attributes ウィンドウ内で、

   - **@PageREF_sym** の **String** 値を入力します（大文字小文字の区別があります）。
   - **PageName** の **Layer** 色を選択します。
   - **FontObj** 値が **SystemH[2.500]W[1.6664]T[0.3750]**であることを確認します。
   - **OK** をクリックして値を確定して **Text Attributes** ウィンドウを閉じます。

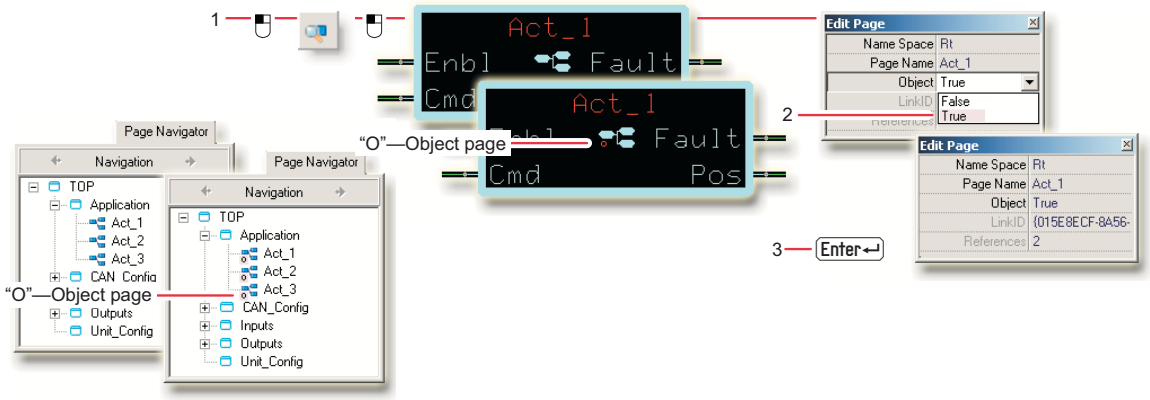


3. リンクシンボルを表示させたい場所に**@PageREF_sym** 文字列の左下隅を配置します。

4. **Page Interface Editor** ウィンドウを抜けます。

リンクページのオブジェクトページへの変換

一般的に、リンクページをオブジェクトページに変換すると、フラッシュメモリ消費が少なく、RAM メモリ消費が若干大きいコンパイルコードが生成されます。



リンクページをオブジェクトページに変換するには、

1. リンクページを Query します。

   Edit Page ウィンドウが表示されます。

2. **Edit Page** ウィンドウ内で、**Object** プロパティを **True** に設定します。

3. Enter を押します。

   - ページには小さい"O"がリンクシンボルの下に表示され、それがオブジェクトページであることを示します。
   - 他の全てのページがオブジェクトページになります。
   - **Page Navigator** タブでは、小さな"O"がページ名のリンクシンボルの下に表示され、それがオブジェクトページであることを示します。

リンクページとオブジェクトページで作業する

リンクページやオブジェクトページで作業するときは以下のことに注意してください。

- リンクやオブジェクト機能を持つプロジェクトを作成するときは、常に最新のハードウェアファイル（*.HWD）を使用してください。

  （通常、リンクページの **Object** プロパティを **True** に設定できない場合は新しいハードウェアファイルが必要です。）

- アロケーションエラーによるコンパイルエラーを避けるため、チェックポイントやメモリーコンポーネントを含むリンクページには **Name Space** プロパティを使用してください。

- リンクページ内では使用できないコンポーネントがあります。これらのコンポーネントのリストは***479***ページの*リンクページで使用できないコンポーネント* を参照してください。

- オブジェクトページ内では使用できないコンポーネントがあります。これらのコンポーネントのリストは***480***ページの*オブジェクトページで使用できないコンポーネント* を参照してください。

- ロジックを含まないオブジェクトページ（**Component** タブの **Page** など）はコンパイラエラーの原因となります。

- 他のページとリンクしていない単独オブジェクトページはコンパイルエラーの原因となります。

  同一のオブジェクトページのうち、一つを残して他を削除した場合にこのエラーが発生します。**Object** プロパティが **True** に設定された単一ページが残ってしまいます。**Object** プロパティは **False** に戻すことはできません。

  このエラーを修正するには、

  1. 問題のあるオブジェクトページを複製します。
  2. 問題のあるオブジェクトページの **Object** プロパティを **True** から **False** に変更します。
  3. 複製したオブジェクトページを削除します。

- PLUS+1 GUIDE プログラムは、リンクページやオブジェクトページの内部での変更を更新するのに時間が掛かります（特にこれらのページが大規模の場合）。

  PLUS+1 プログラムの応答時間を改善するには、できるだけ単一ソースページ上で作業することです。そして、ソースページを使ってリンクページやオブジェクトページを作成します。

- PLUS+1 GUIDE プログラムはリンクページやオブジェクトページを作成するときに一時ファイルを作成します。プロジェクトをパックするときにこれらのファイルは削除されません。

  これらのファイルを除去するには、プロジェクトをパックしてからプロジェクトフォルダに入り、一時ファイルを削除してください。

リンクページで使用できないコンポーネント

これらのコンポーネントは、リンクページ内で複製されるとコンパイルエラーを生じさせます。

これらのコンポーネントの複製コピーは、リンクページを含むか含まないかに拠らずどのようなアプリケーションであってもコンパイラエラーを生じさせます。このような場合には、ページのリンクを解消してもコンパイラエラーを解消することはできません。

リンクページで使用できないコンポーネント

コンポーネントのカテゴリ	コンポーネント名	コメント
Connection	Simple Checkpoint	
	Initialize Hardware Output	
	Hardware Output	
	Hardware Input/Output Typed	
	Read Output from Hardware Typed	
	Non-Volatile Memory Input Typed	
	Non-Volatile Memory Input/Output	
Module	Module Bus Output	
	Module Output	
Array	Array Constant from Binary File	
Access	Access App Log Enable	
	Disable Raw Applog Data Readout	
	Accessrights App Log Statistics	
	Accessrights App Log Errors	
	Accessrights App Log Others	
	Accessrights History	
	Accessrights Read	
	Accessrights Write	

オブジェクトページで使用できないコンポーネント

オブジェクトページ内でこれらのコンポーネントを使用するとコンパイラエラーを生じます。

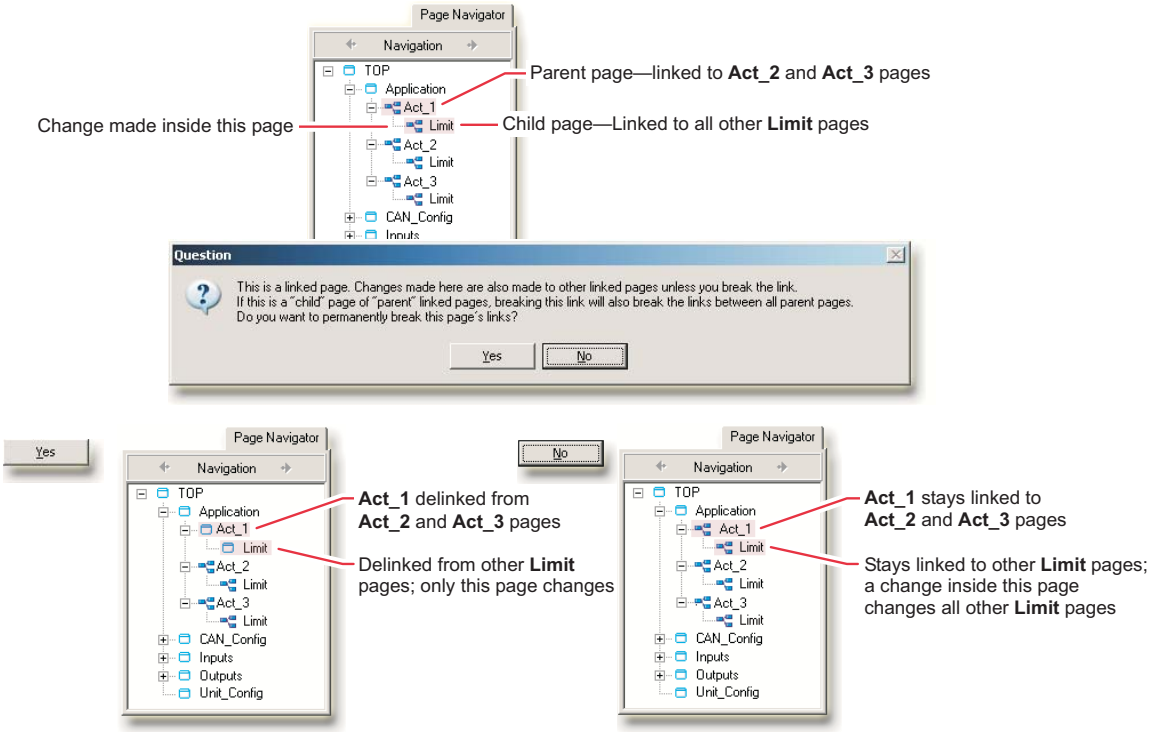
オブジェクトページで使用できないコンポーネント

コンポーネントのカテゴリ	コンポーネント名	コメント
Constant	Multi-character	データ型がINTまたはBOOLの場合のみ可能
Switch,Counter, Memory	Value Initialization	
Array	Array Constant from File	
	Array Constant from Binary File	
Connection	Simple Checkpoint	
	Advanced Checkpoint	
	Set Value	
	Set Pulse	
	Initialize Hardware Output	
	Hardware Input Typed	
	Hardware Input	
	Hardware Output	
	Hardware Input/Output Typed	
	Hardware Input/Output	
	Read Output from Hardware Typed	
	Read Output from Hardware	
	Transmit CAN	
	Receive CAN with Filter	
	Receive CAN with ID Mask	
	Receive CAN Basic	
Module	Module Input Typed	
	Module Input	
	Module Output	
	Configurable Module Connection	
	Module Bus Input	
	Module Bus Output	
	Call Module	
Manage	Generate Compiler Error on Range	
	Generate Compiler Error on Not Constant	
	Generate Compiler Error on Type	
	Predefine Type	
Access	Access App Log Enable	
	Disable Raw Applog Data Readout	

オブジェクトページで使用できないコンポーネント

コンポーネントのカテゴリ	コンポーネント名	コメント
	Accessrights App Log Statistics	
	Accessrights App Log Errors	
	Accessrights App Log Others	
	Accessrights History	
	Accessrights Read	
	Accessrights Write	
Read-only Parameter	Open Parameter Set	
	Close Parameter Set	
	Read-only Parameter Input with Namespace	
	Read-only Parameter Input	
	Read-only Parameter Input Typed	
Display	Define Window	
	Line	
	Graphic	
	Text Label	
	Graphic Label	
	String	
	String with 4 Variables	
	String with 8 Variables	
	Basic String with 4 Variables	
	Basic String with 8 Variables	
	Open Text Set	
	Close Text Set	
	Select Language	
	Define Areas Page	
	Define Screen Page	
Application Log	Language Definition Input	
	Define Application Log Areas Page	
	Define Application Log Page	

リンクページ内での変更



リンクページ内で変更する場合は、**Question** ウィンドウが以下の内容を表示します。

> **This is a linked page. Changes made here are also made to other linked pages unless you break the link. If this is a "child" page of "parent" linked pages, breaking this link will also break the links between all parent pages.**
>
> **Do you want to permanently break this page's links?**
>
> （訳： これはリンクページです。ここで変更した内容はリンクを切らない限り他のリンクページにも反映されます。もしこれがリンクされた親ページの子ページである場合、リンクを切ると全ての親ページのリンクを切ることになります。
>
> 恒久的にページリンクを切りますか？）

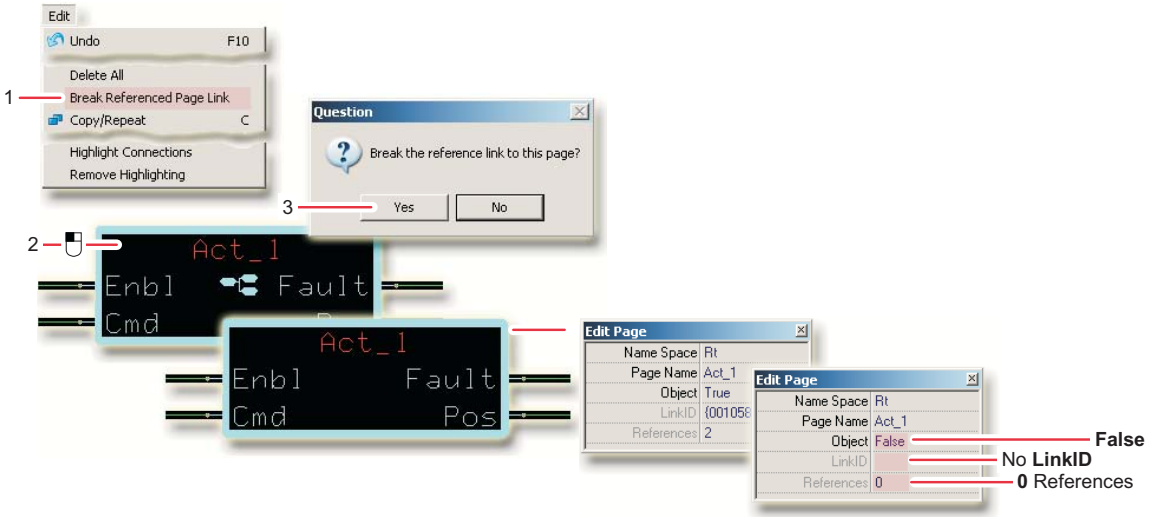
No を選択した場合、

- リンクページ内の変更は他の全てのリンクページにも同じように反映されます。
- 親子ページのリンク関係も保たれます。

Yes を選択した場合、

- リンクページはリンクが解消されます
- 変更点は現在のページのみに適用されます.
- リンクを解消したページが親リンクページの子ページの場合、親ページもそのリンクが解消されます。

ページ間のリンクを切る



1. **Edit** メニューで **Break Referenced Page Link** コマンドをクリックします。
2. リンクを切りたいページのポートの周りをドラッグします。

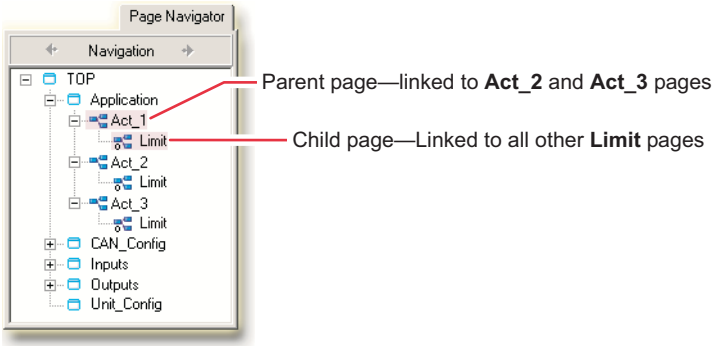
   Question ウィンドウが表示されます。
3. **Question** ウィンドウで **Yes** をクリックします。

   Yes をクリックすると、ページのリンクは切れ、リンクシンボルも表示されなくなります。しかしこのページ内のリンク子ページはリンクが保たれています。

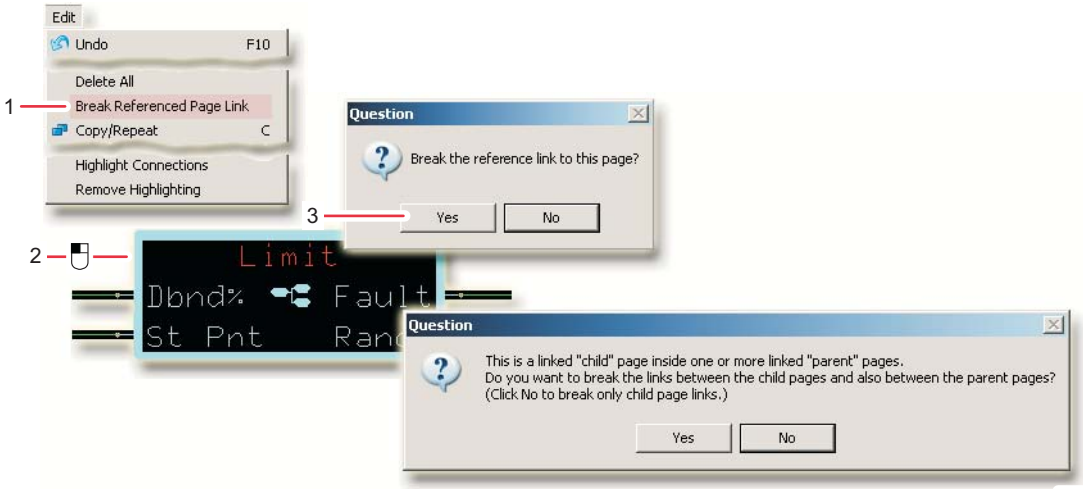
   リンクが解消されたページの **Edit Page** ウィンドウには以下のような表示がされます。

   - **Object** ステータス＝**False**
   - **LinkID** なし
   - **References** の数値＝**0**

子ページ間のリンクを切る



「子」ページとはリンクされた「親」ページの中にあるリンクページです。



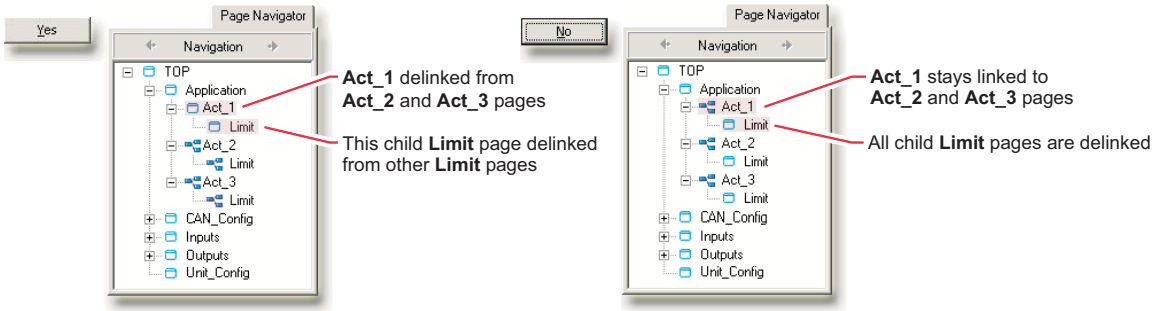
1. **Edit** メニューで **Break Referenced Page Link** コマンドをクリックします。
2. リンクを切りたい子ページのポートの周りをドラッグします。

   Question ウィンドウが表示されます。
3. **Question** ウィンドウで **Yes** をクリックします。

   Yes をクリックすると、2 番目の **Question** ウィンドウが以下の内容を表示します。

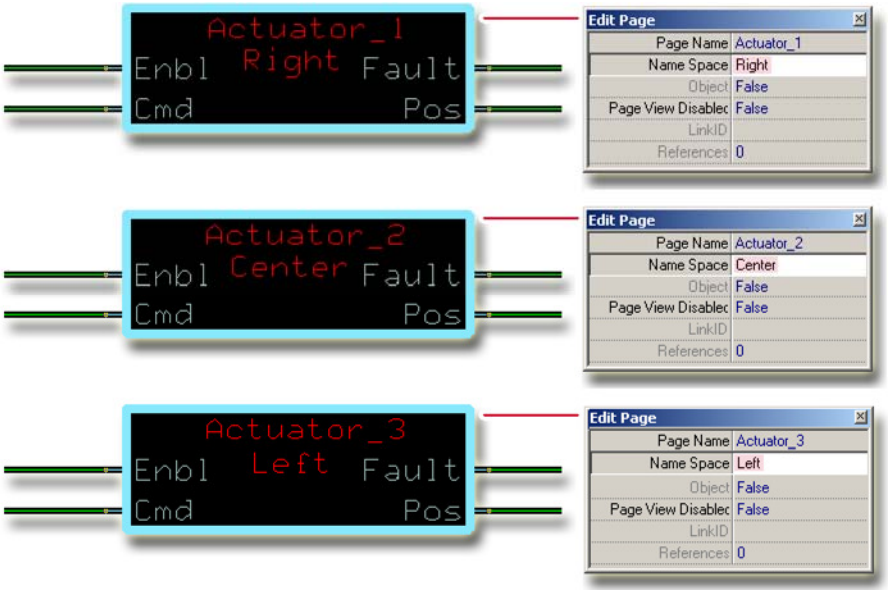
   This is a linked “child” page inside one or more linked “parent” pages.
   Do you want to break the links between the child pages and also between the parent pages?
   (Click No to break only child page links.)

   （訳：これは一つあるいはそれ以上のリンクされた親ページ内のリンク子ページです。子ページおよび親ページ間のリンクを切りますか？
   （NO をクリックすると子ページのリンクのみ切ります。））



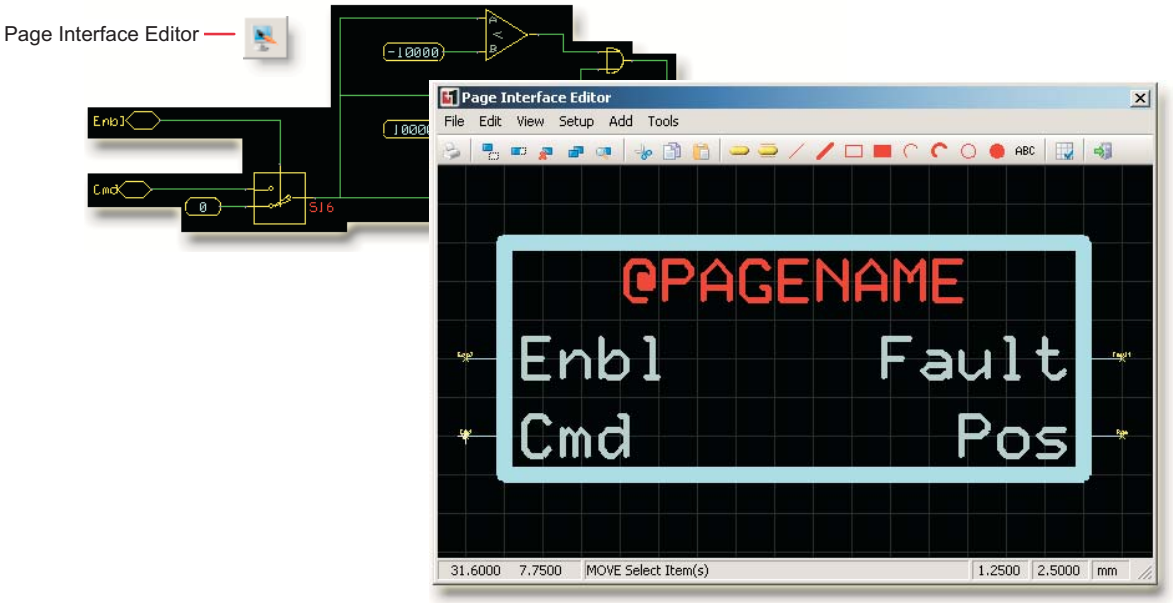
- **Yes** を選択した場合、
  - 親ページはリンクが解消されます
  - 子ページもリンクが解消されます
- **No** を選択した場合、
  - 親ページはリンクが保たれます
  - 全ての子ページはリンクが解消されます

ページのName Spaceの表示

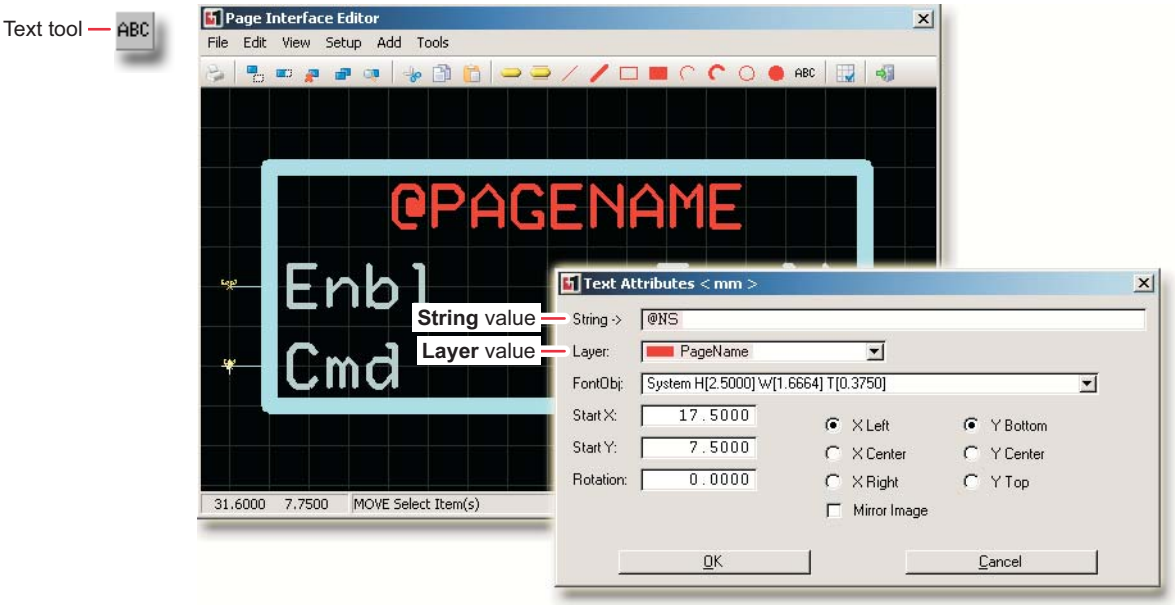


ネームスペース機能を使用したコンポーネント（**Advanced Checkpoint with Namespace** や **Read-only Parameter with Namespace** など）を含むページの **Name Space** を表示することで、アプリケーションが見やすくなり、メンテナンスもしやすくなります。

以下の手順で、ページの **Name Space** 値を表示させることができます。



1. 表示したい **Name Space** 値を持つページのトップビューを編集するため、**Page Interface Editor** に入ります。



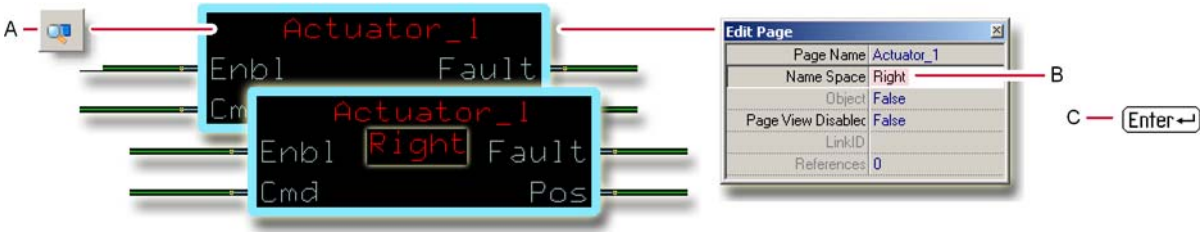
2. Text ツールをクリックして **Text Attributes** ウィンドウを表示します。

   Text Attributes ウィンド内で、

   - **@NS** の **String** 値を入力します（大文字小文字を区別）
   - **Layer** の色を選択します（**PageName** が標準の色です）。
   - **FontObj** を適切な値に選択します。

   OK をクリックしてこれらの値を確定し、**Text Attributes** ウィンドウを閉じます。

3. **Page Interface Editor** を抜けてページのトップビューに戻ります。



4. ページの **Name Space** 値を表示します。

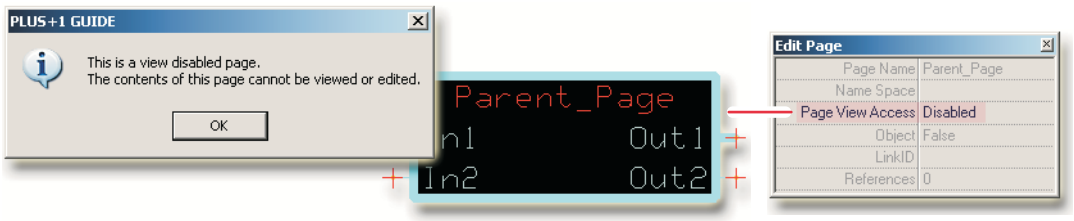
   A. 表示したいページの **Name Space** 値を Query します。

   Edit Page ウィンドウが表示されます。

   B. **Edit Page** ウィンドウ内で、ページの **Name Space** 値を入力します。

   C. Enter を押します。

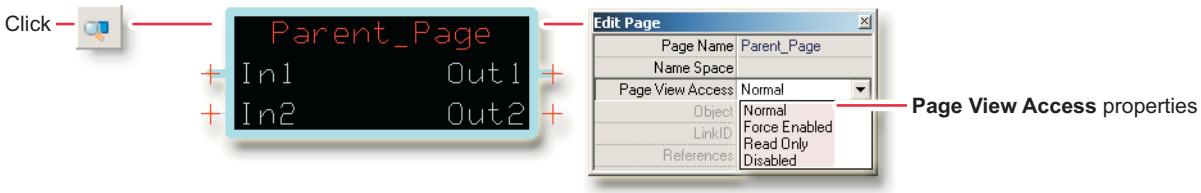
ページビューの無効化（ロック）



Page View Access プロパティを使用してページへのアクセスを制限します。

Page View Access を変更した場合、これを元に戻すことはできません。アクセスプロパティを設定するときに失敗すると複数のページにアクセスできなくなります。

アクセスプロパティを変更する前に、**File** メニューの **Export** ブロックコマンドを使ってページのバックアップを取ることを強く推奨します。



各ページはデフォルトで **Normal** の **Page View Access** プロパティを持っています。**Edit Page** ウィンドウを使用してこのアクセス値を **Force Enabled**、**Read-only**、または **Disabled** に変更します。

Normal → Force Enabled → Read-only → Disabled
Force Enabled → Read-only → Disabled
Read-only → Disabled
Disabled

Changes cannot be reversed

ページアクセスプロパティの変更は恒久的で、元に戻すことはできません。

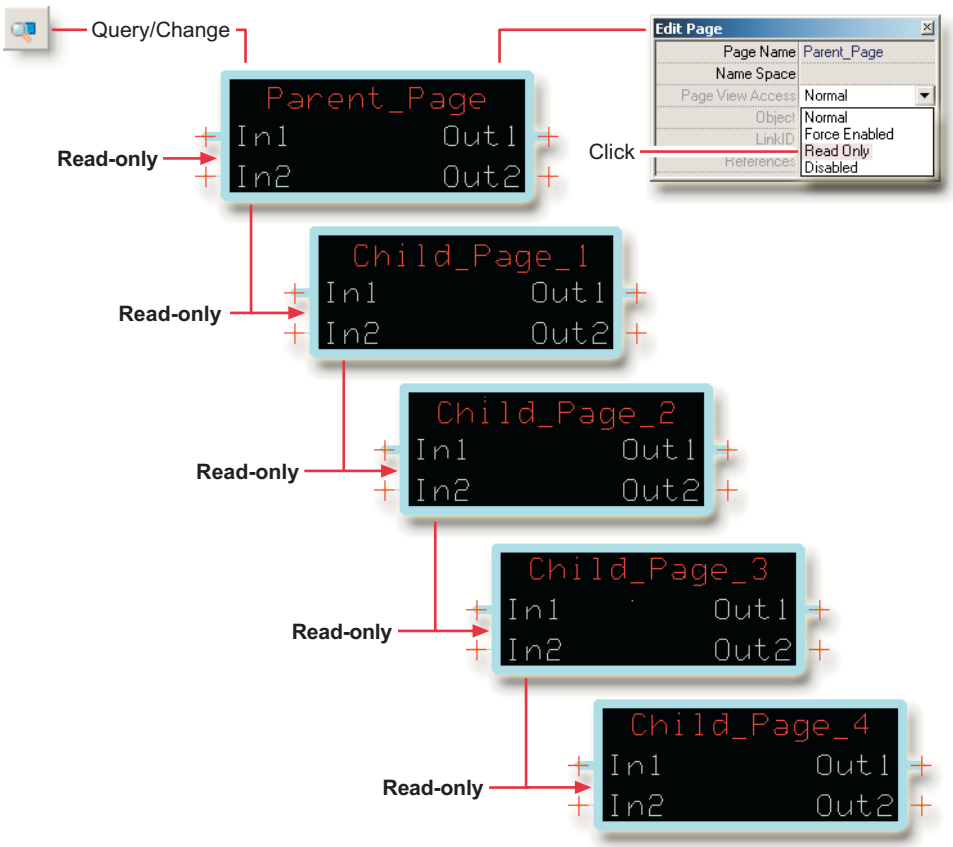
Page View Access プロパティ

プロパティ	説明
Normal	ページ内容を見たり変更することができます。全てのページのデフォルト状態はこれです。 **Normal** アクセスプロパティを **Force Enabled**、**Read-only**、または **Disabled** に変更することができます。
Force Enabled	親ページの **Disabled** や **Read-only** アクセスプロパティを子ページに引き継がないようにします。ページの内容は見たり変更することができます。 親ページのアクセスプロパティを **Read-only** や **Disabled** に変更する前に子ページのアクセスプロパティを **Force Enabled** に設定してください。

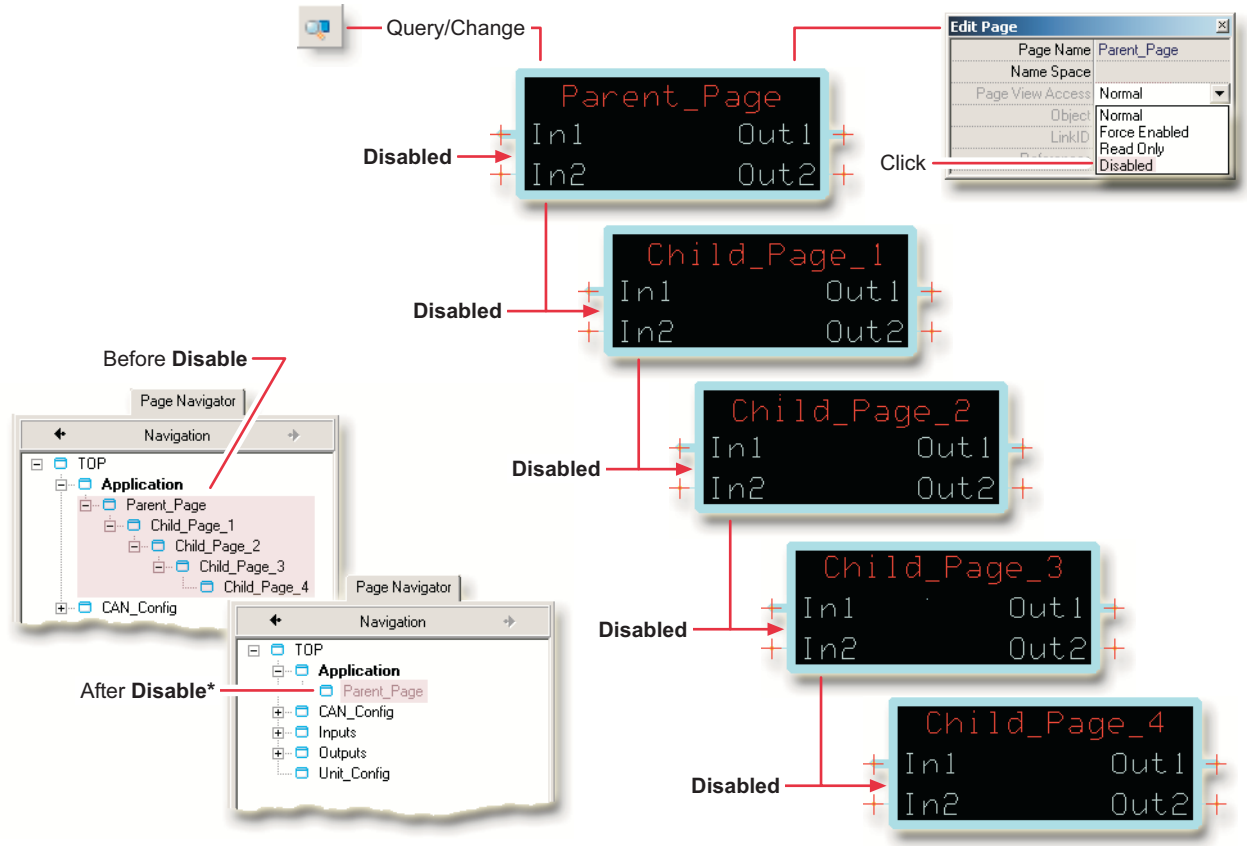
Page View Access プロパティ

プロパティ	説明
Read-only	ページ内容は見ることはできますが変更することはできません。 親ページのアクセスプロパティを **Read-only** に変更すると、その子ページの **Normal** アクセスプロパティも **Read-only** に変更されます。
Disabled	ページ内容を見ることはできません。 親ページのアクセスプロパティを **Disabled** に変更すると、その子ページの **Normal** や **Read-only** アクセスプロパティも **Disabled** に変更されます。

アクセスプロパティ

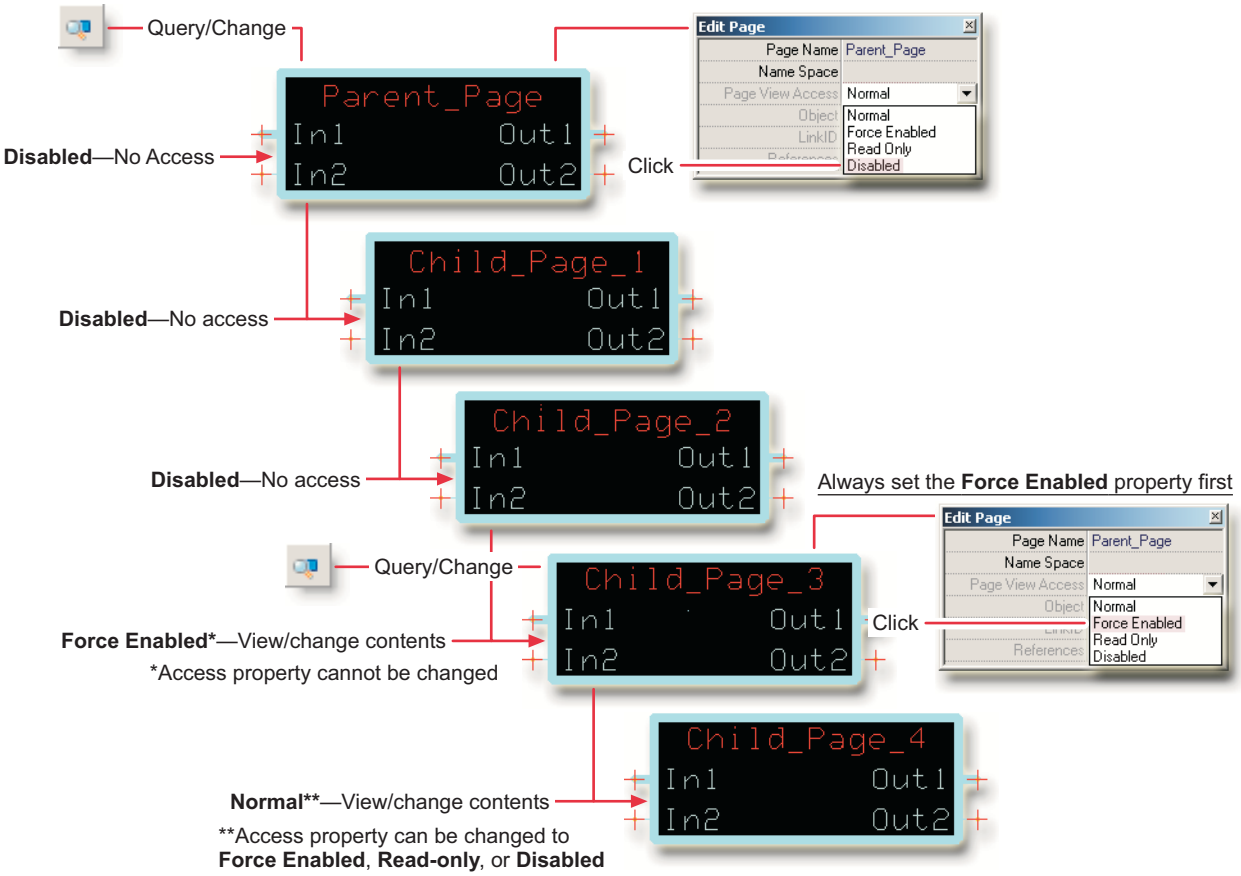


- 親ページの **Page View Access** プロパティを **Read-only** に変更すると、全ての子ページのアクセスプロパティも **Read-only** に設定されます。
- **Read-only** ページの内容は見ることはできますが編集することはできません。
- **Normal** アクセスプロパティを持つリンクページも **Read-only** に変更されます。
- **Disabled** の子ページも disabled のまま保たれます。
- アクセスプロパティを一度 **Read-only** に設定すると二度と **Normal** に戻すことはできません。.
- **Read-only** アクセスプロパティは **Disabled** に変更することができます。



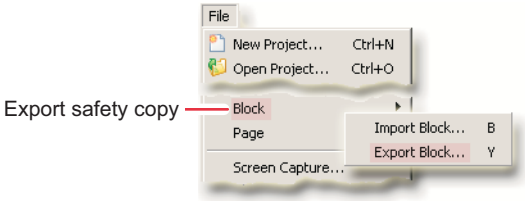
*Set how the the Navigation tab displays disabled pages in the Options window.

- 親ページの **Page View Access** プロパティを **Disabled** に変更すると、全ての子ページのアクセスプロパティが **Disabled** に設定されます。
- **Disabled** ページの内容は見ることができません。
- **Normal** または **Read-only** アクセスプロパティのリンクページも **Disabled** に変更されます。
- アクセスプロパティをいったん **Disabled** 設定すると、二度と **Read-only**、**Force Enabled**、または **Normal** に戻すことはできません。

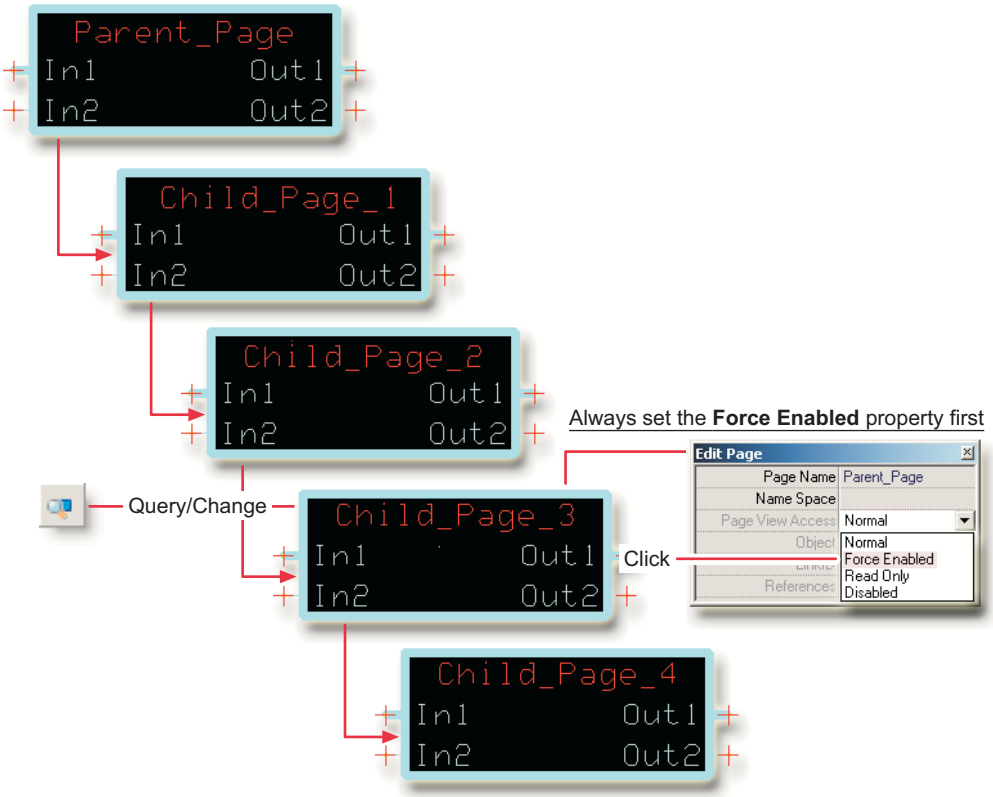


- 子ページの **Force Enabled** アクセスプロパティは、その親ページの **Disabled** または **Read-only** アクセスプロパティを引き継がないようにします。
- **Force Enabled** ページ内の子ページのアクセスプロパティは変わりません。
- 親ページのアクセスプロパティを **Read-only** または **Disabled** に変更する前に、子ページのアクセスプロパティを **Force Enabled** に設定してください。
- **Force Enabled** ページの内容は見ることも編集することもできます。
- **Normal** アクセスプロパティのリンクページも **Force Enabled** に変更されます。
- アクセスプロパティをいったん **Force Enabled** に設定すると、二度と **Normal** に戻すことはできません。.
- **Force Enabled** アクセスプロパティは **Read-only** または **Disabled** に変更することができます。

Page View Access プロパティを変更するには、以下の手順を実行します。

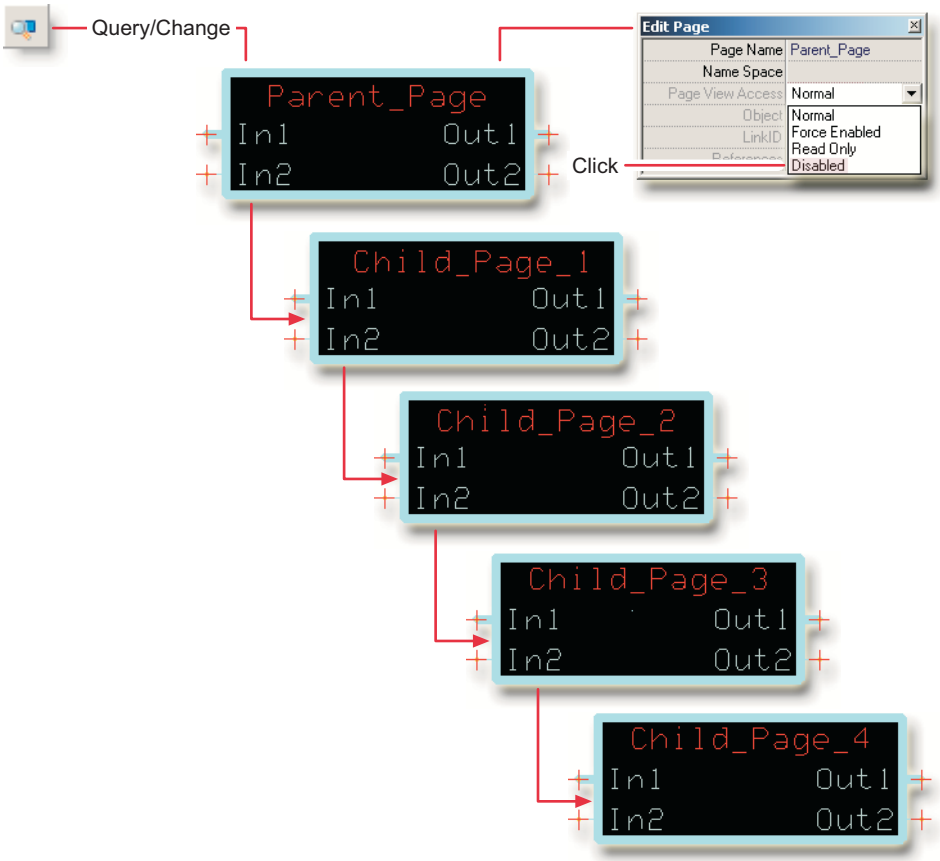


1. アクセスプロパティを変更したいページの安全コピーをエクスポートします。



2. 子ページの読み取り/書き込みアクセスを確保したい場合は、子ページに行ってそのアクセスプロパティを **Force Enabled** に設定してください。

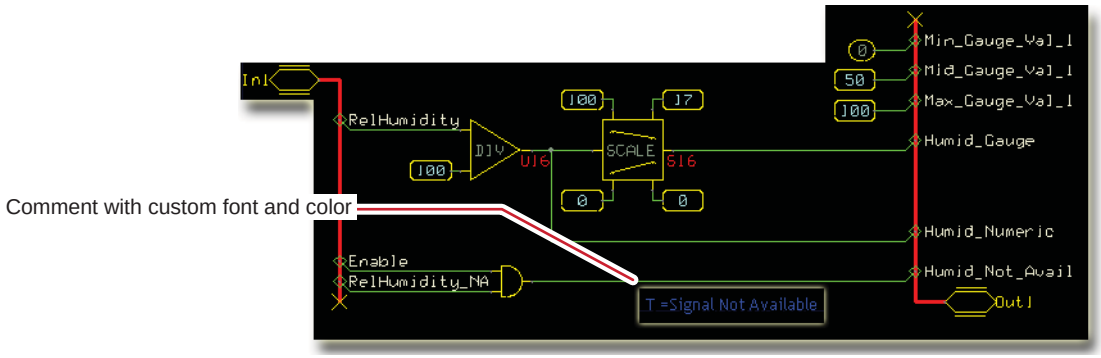
   必要なければ次のステップに進みます。



3. アクセスプロパティを変更したいページに行きます。

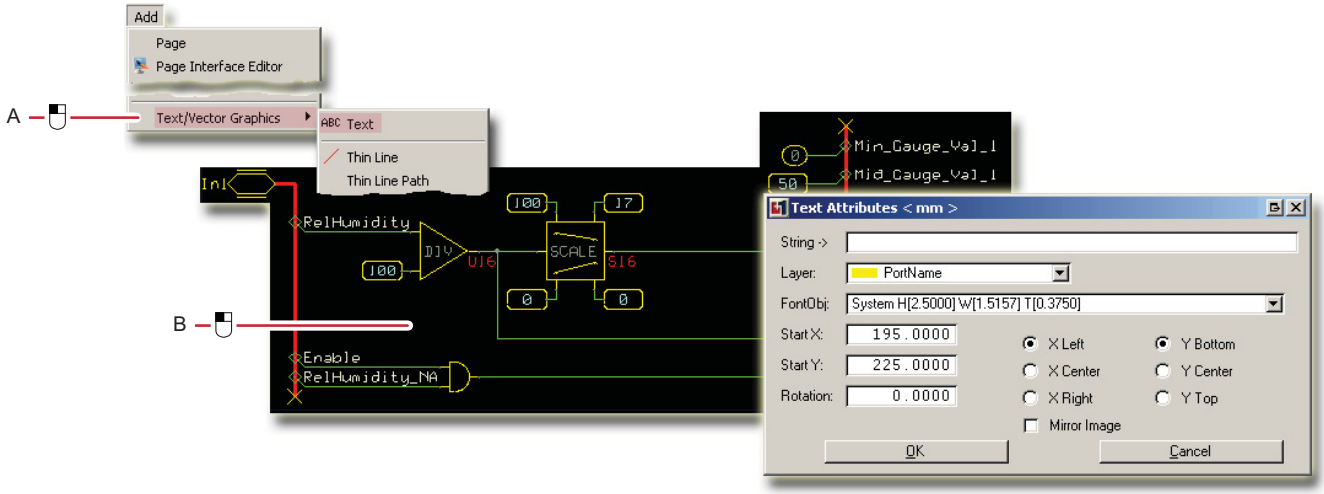
   ページを Query し、アクセスプロパティを **Read-only** または **Disabled** に変更します。

コメントのフォントと色をカスタマイズする

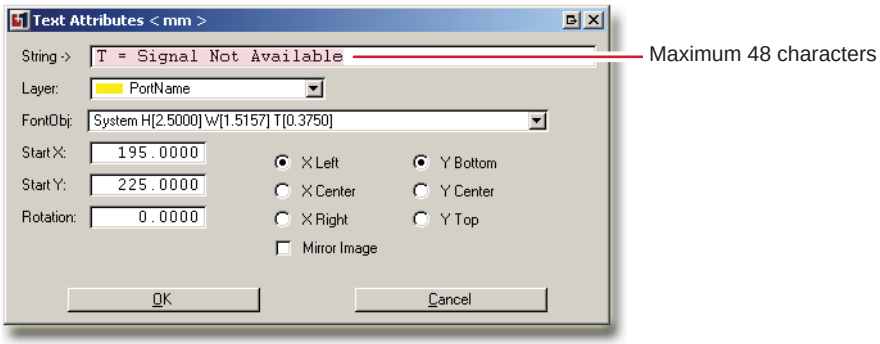


Text Attributes ウィンドウを使用して、アプリケーション内のコメント文字列のフォント、サイズ、または色をカスタマイズします。

> 正しいフォントが PC にインストールされていなければ正しく表示されません。アプリケーションを複数の PC で編集する場合、Microsoft OS に共通のフォント（Arial や Trebuchet など）を選択するように心がけてください。

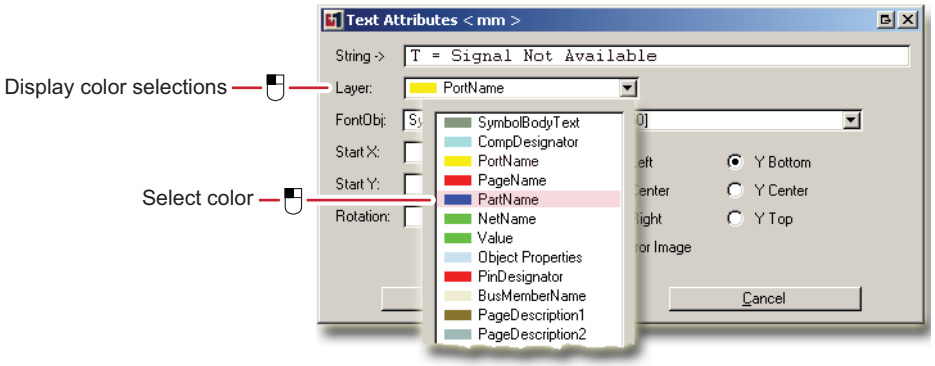


1. **Text Attributes** ウィンドウを表示します。
   A. メニューバー > **Add** メニュー > **Text/Vector Graphics** > **Text** をクリックします。
   B. 図面エリアをクリックします。



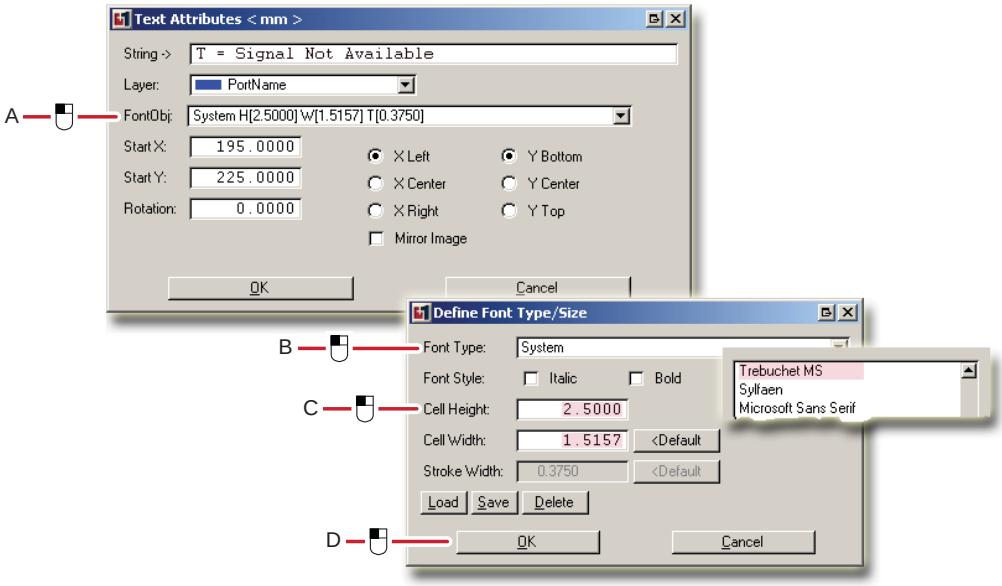
2. **Text Attributes** ウィンドウの **String** テキストボックスにコメントを入力します。

   一つのコメントは最大 48 文字まで可能です。



3. 必要に応じてコメント文字列の色を変更します。

   Layer プルダウンメニューで文字色を選択してクリックします。



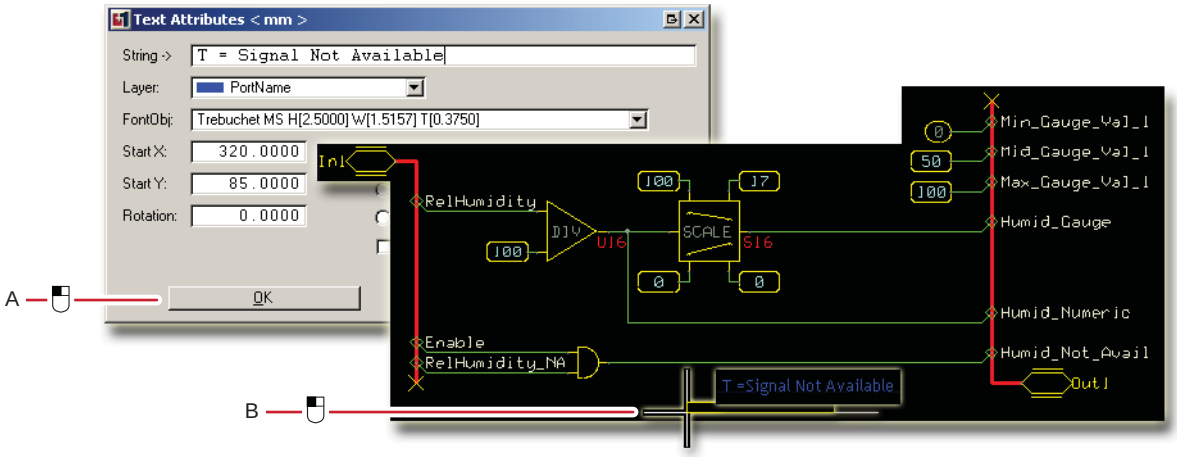
4. 必要に応じて、コメント文字列のフォントとフォントサイズを変更します。

   A. **Text Attributes** ウィンドウで **FontObj** プルダウンメニューをクリックして **Define Font Type/Size** ウィンドウを表示します。

   B. **Define Font Type/Size** ウィンドウで、**Font Type** プルダウンメニューをクリックしてコメント文字列の新しいフォントを選択します。

   C. **Define Font Type/Size** ウィンドウで、**Cell Height** および **Cell Width** テキストボックスに新しい値を入力してコメント文字列のフォントサイズを変更します。

   D. **OK** をクリックして **Define Font Type/Size** ウィンドウを閉じます。



5. コメントを配置します。

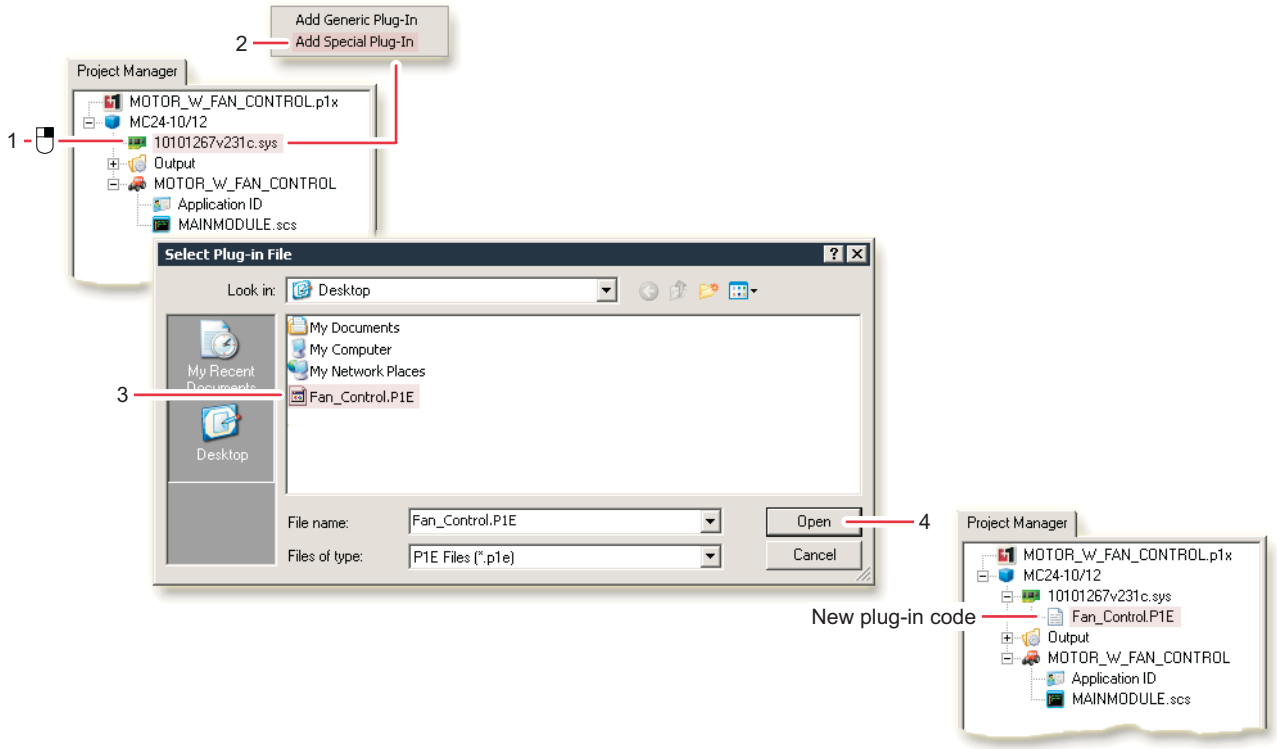
   A. **OK** をクリックして **Text Attributes** ウィンドウを閉じます。

   B. 図面エリアの所望の位置にコメントを移動し、もう一度クリックしてコメントを配置します。

プラグインコードの追加

プラグインコードによりアプリケーションの機能を拡張することができます。

この機能のすべてについては将来のマニュアル改訂で追加されます。



この手順ではプラグインコード(プレコンパイルコード)をアプリケーションに追加する方法を記述します。

1. **Project Manager** タブで、システムファイルを右クリックしてポップアップメニューを表示します。
2. ポップアップメニューで Add Generic Plug-in または Add Special Plug-in を選択して **Select Plug-in File** ウィンドウを表示します。
3. **Select Plug-in File** ウィンドウでプラグインファイルをブラウズして選択します。
4. **Select Plug-in File** ウィンドウで **Open** をクリックします。

   Project Manager に新しいプラグインコードが表示されます。

スクリーンエディタの手引き

この章は PLUS+1 GUIDE スクリーンエディタの手引きをまとめたものです。

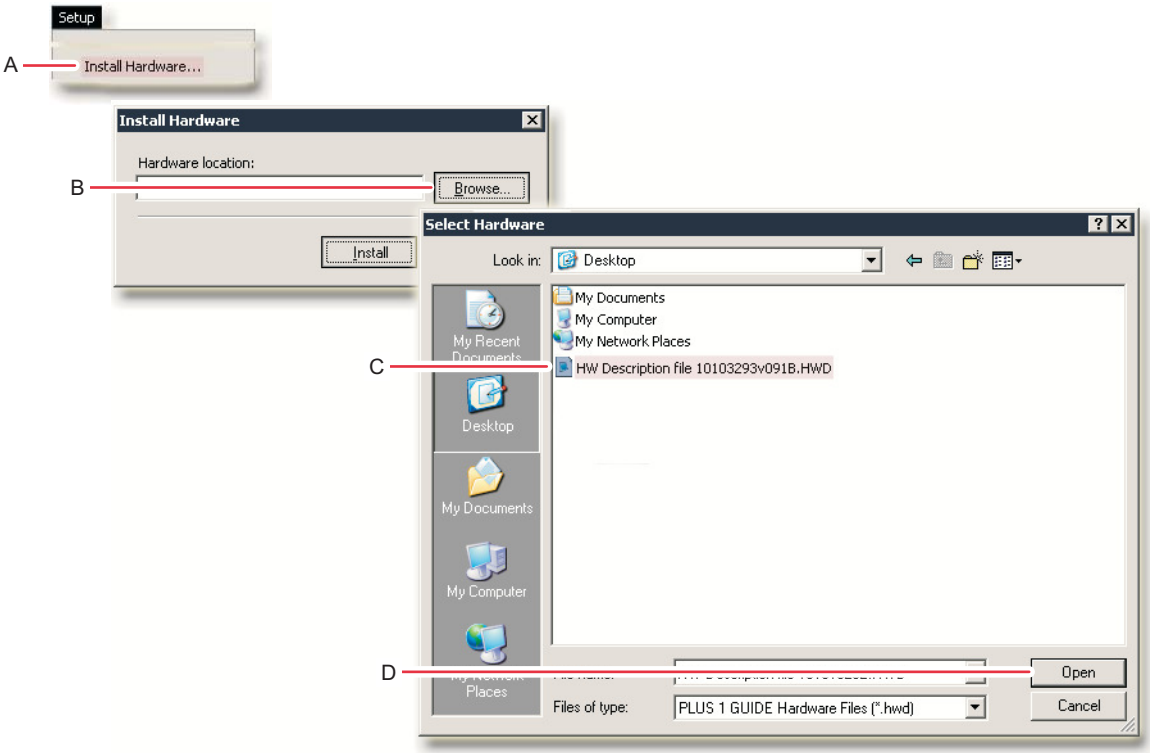
この章の内容

スクリーンエディタの手引き 501
スクリーンエディタプロジェクトの開始 502
Define Areas ページ/ハードウェアポートの割当 505
Define Areas ページ/追加スクリーン・エリアの割当 506
Define Screen ページ/スクリーン・エリアの名前の変更 507
Define Screen ページ/スクリーン・エリアの削除 508
Define Screen ページ/スクリーン・エリアのプロパティの変更 509
Define Areas ページ/信号により有効化されたスクリーン・エリアの初期化 511
Define Screen ページ/スクリーン・エリアの割当 512
Define Screen ページ/スクリーン・エリアの削除 513
Define Screen ページ/バーグラフの作成 514
Define Screen ページ/テキストグループの作成 515
Define Screen ページ/デフォルト言語設定の使用 516
デフォルト言語の変更方法 517
Define Screen ページ/言語の切換 518
Define Screen Page/Import Translated Text into an Application 519
Export a CSV File with the Text that Needs Translation 519
Translate the Text 521
Import a CSV File with the Translated Text 523
About the Language Order 524
Define Screen ページ/データ値の表示 525
Define Screen ページ/データ値のフォーマット 526
Define Screen ページ/画像リストの表示 528
Define Screen ページ/テキストリストの表示 533
Define Screen ページ/アプリケーションフォントのインストール 538
Define Screen ページ / 文字列のフォントプロパティの切り替え 544
Define Screen ページ / 特殊キャラクタの表示 548
キーストロークコードによる方法 549
キャラクタコードによる方法 552
Define Screen ページ / 非ローマンアルファベットの表示 555
Define Screen ページ/動画表示 560
Define Screen ページ/画面印刷 562

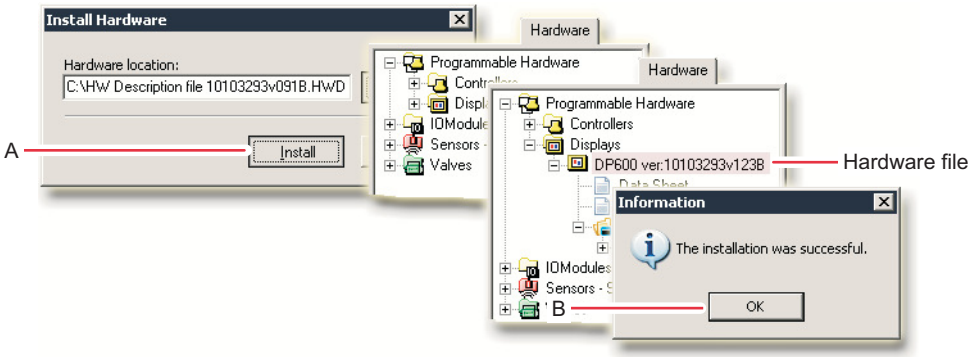
スクリーンエディタプロジェクトの開始

スクリーンエディタプロジェクトは以下の手順で開始します。このプロジェクトでは、PLUS+1 ディスプレイ用のアプリケーションを作成します。

1. パソコンに PLUS+1 GUIDE アプリケーションの最新バージョン（2.2.4 以降）をインストールします。
2. PLUS+1 アプリケーションを起動します。
3. ディスプレイ用のハードウェアファイルをインストール済みのときは、ステップ 6 に進んでください。



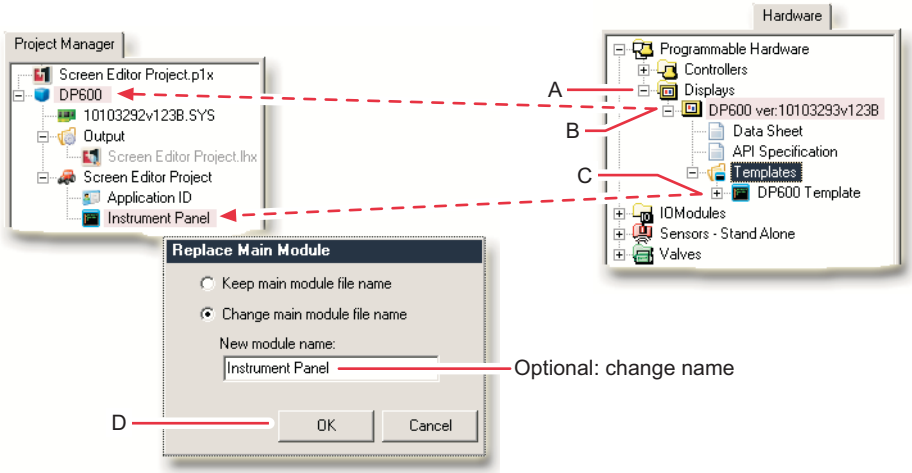
4. アプリケーションを作成するディスプレイ用のハードウェアファイルをインストールします。
   A. **Setup** メニューの **Install Hardware** をクリックします。
   B. 開いた **Install Hardware** ウィンドウで、**Browse** をクリックします。
   C. 開いた **Select Hardware** ウィンドウで、ディスプレイ用のハードウェアファイルをクリックします。
   D. **Select Hardware** ウィンドウで、**Open** をクリックします。



5. ディスプレイ用のハードウェアファイルをインストールします。

   A. **Install Hardware** ウィンドウで、**Install** をクリックします。選択したハードウェアファイルがインストールされ、**Hardware** タブに選択したハードウェアファイルが表示されます。

   B. **OK** をクリックして、**Information** ウィンドウを閉じます。



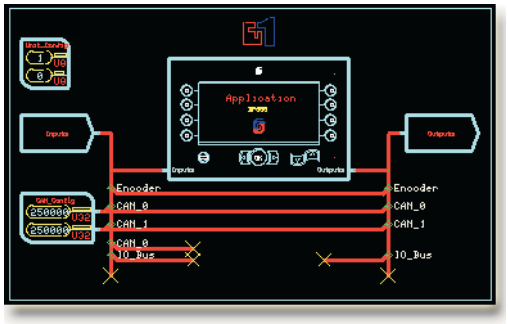
6. ディスプレイ用のハードウェアファイルとテンプレートを、**Hardware** タブから **Project Manager** タブにクリック＆ドラッグします。

   A. **Hardware** タブで、**+** をクリックしてディスプレイのツリーに全項目を表示します。

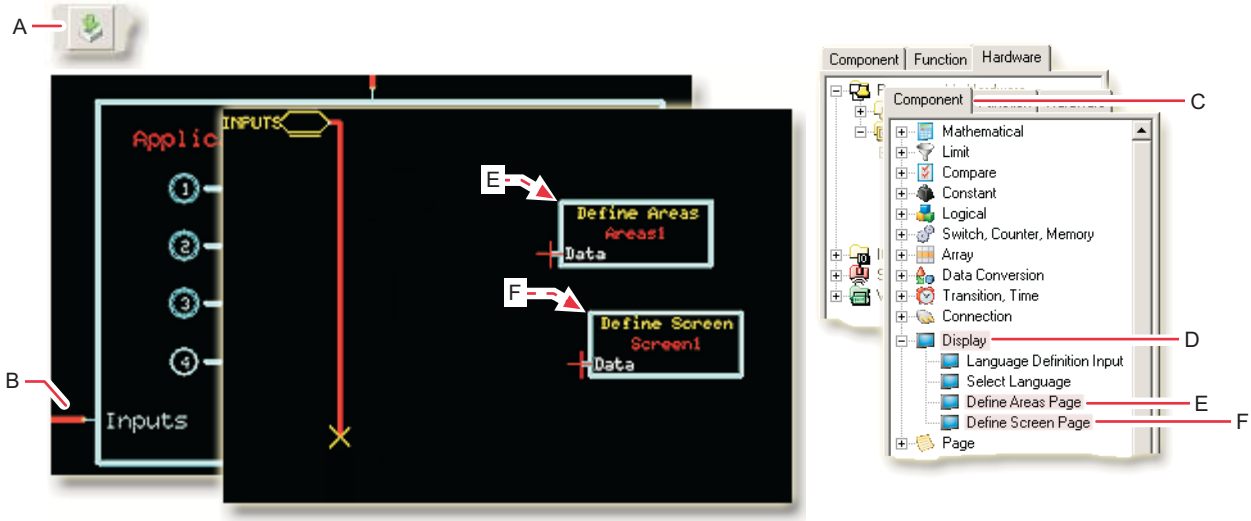
   B. ディスプレイ用のハードウェアファイルを、**Hardware** タブから **Project Manager** タブにドラッグします。

   C. ディスプレイ用のテンプレートを、**Hardware** タブから **Project Manager** タブにドラッグします。必要に応じて、**Replace Main Module** ウィンドウでモジュールの名前を変更します。

   D. **Replace Main Module** ウィンドウで、**OK** をクリックします。



これでディスプレイ用アプリケーションを作成するテンプレートが図面エリアに表示されます。（実際のテンプレートはこれとは少し異なる場合があります）



7. テンプレートの **Application** ページに入って、スクリーンエディタページを追加します。

   このステップで **Define Areas** ページと **Define Screen** ページを追加します。スクリーンエディタを使用するディスプレイアプリケーションには、必ずこの 2 つのページが必要です。

   ディスプレイ表示の制御ロジックを入れるのに便利な **Basic Page** があります。このページは後で追加できます。

   通常はこれらのページに **Inputs** 信号を引き込むバスも追加します。*309* ページの図*スクリーンエディタのエレメント*は全て結線済みの例です。

   A. ツールバーで、**Enter Page**（ページに入る）ボタンをクリックします。

   B. **Application** ページのポートをドラッグして、このページに入ります。

   C. **Component** タブをクリックして表示します。

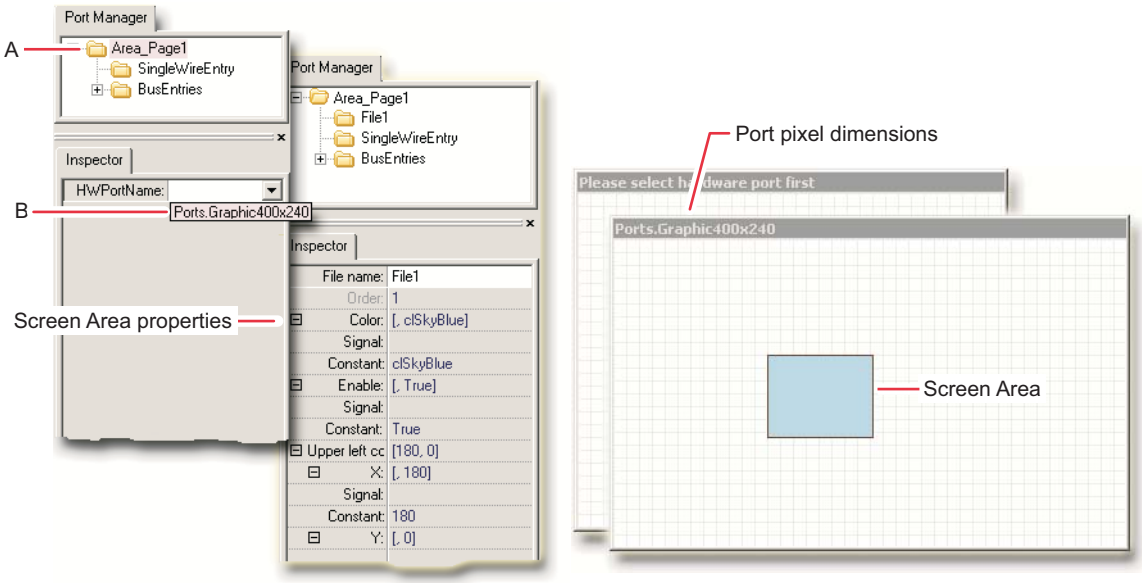
   D. **Component** タブで、クリックして全部の **Display** コンポーネントを表示します。

   E. **Define Areas** ページを **Application** ページにドラッグします。

   F. **Define Screen** ページを **Application** ページにドラッグします。

Define Areasページ/ハードウェアポートの割当

1. **Define Areas** ページに入ります。



2. ハードウェアポートを割当てます。

   A. **Port Manager** タブで、**Area_Page1** をクリックします。

   B. **Inspector** タブで、**HWPortName** をクリックして選択します。

   スクリーンエディタがスクリーン･エリアをレイアウトペインに追加します。

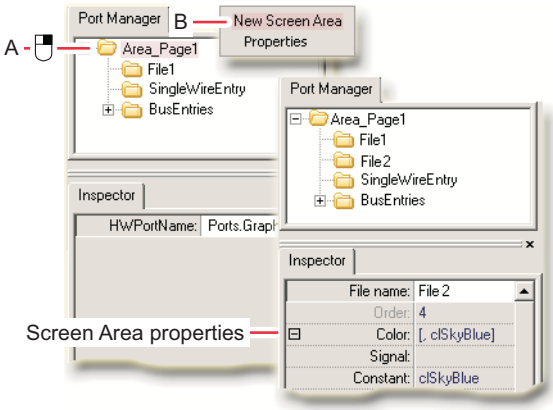
   スクリーンエディタは **HWPortName** の画素寸法に合うようにレイアウトペインのサイズを変更します。

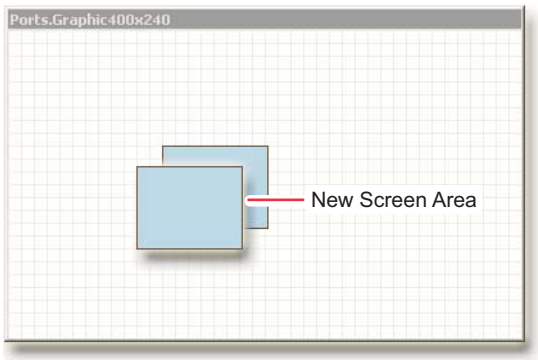
   Inspector タブは **File name** を含むスクリーン･エリアのプロパティを表示します。

Define Areasページ/追加スクリーン・エリアの割当

ハードウェアポートを割当てると（*504*ページ*Define Areas Page/Assign a Hardware Port* 参照）、自動的に最初のスクリーン・エリアが割当てられます。追加スクリーン・エリアの割当ては次の手順で行なってください。

1. **Define Areas** ページに入ります。





2. スクリーン・エリアを追加します。

   A. **Port Manager** タブで、**Area_Page1** を右クリックします。

   B. 開いたポップアップメニューで、**New Screen Area** をクリックします。

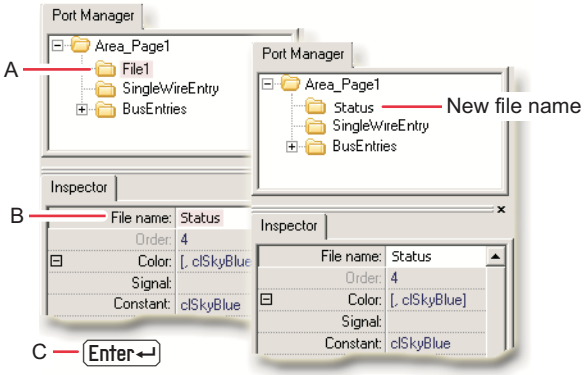
   レイアウトペインに新たなスクリーン・エリアが表示されます。

   Inspector タブは **File name** を含むスクリーン・エリアのプロパティを表示します。

   さらにスクリーン・エリアを追加するときは、この手順を繰返します。

Define Screenページ/スクリーン・エリアの名前の変更

1. **Define Areas** ページに入ります。



2. スクリーン・エリアの名前を変更します。

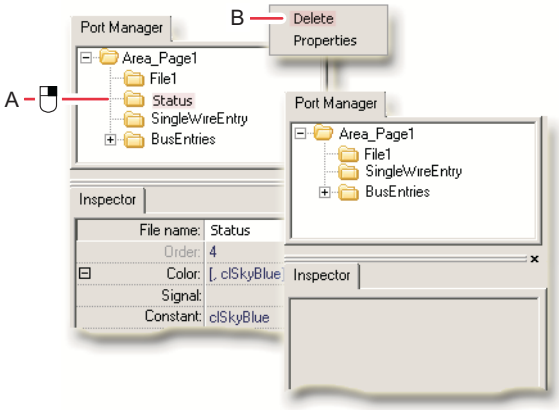
   A. **Port Manager** タブで、変更したいスクリーン・エリア名をクリックします。

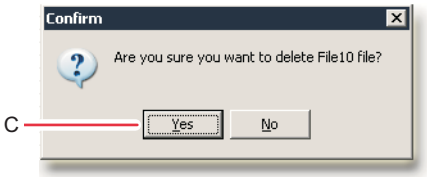
   B. **Inspector** タブで、新しい **File name** をタイプ入力します。

   C. Enterを押します。

Define Screenページ/スクリーン・エリアの削除

1. **Define Areas** ページに入ります。





2. スクリーン・エリアを削除します。

   A. **PortManager** タブで、削除したいスクリーン・エリアを右クリックします。

   B. 開いたポップアップメニューで、**Delete** をクリックします。

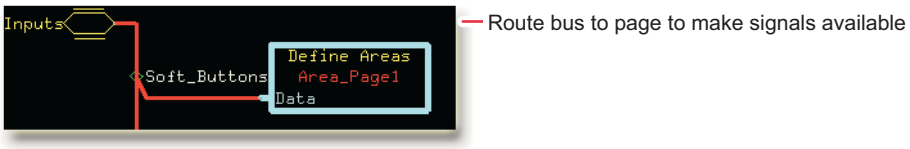
   C. 開いた **Confirm** ウィンドウで、**Yes** をクリックします。

   スクリーン・エリアがレイアウトペインから消えます。

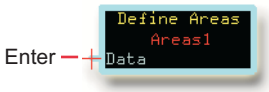
   他のスクリーン・エリアを削除するときは、この手順を繰返します。

   最後の画面は削除できません。常にひとつのスクリーン・エリアがハードウェアポートに割当てられたままになります。

Define Screenページ/スクリーン・エリアのプロパティの変更

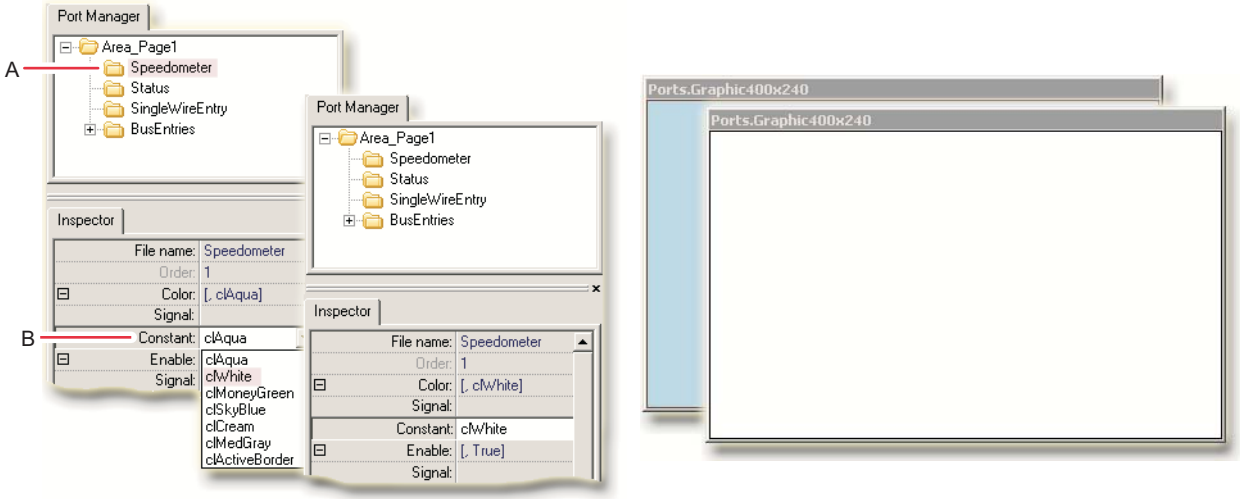


1. 信号でプロパティを制御したいときは、その信号を含むバスを **Define Areas** ページに引き込んで使用できるようにしてください。



2. **Define Areas** ページに入ります。
3. 定数のスクリーン・エリアのプロパティを変更するときは、ステップ 4 を実行します。

   スクリーン・エリアのプロパティを信号で制御するときは、ステップ 5 を実行します。

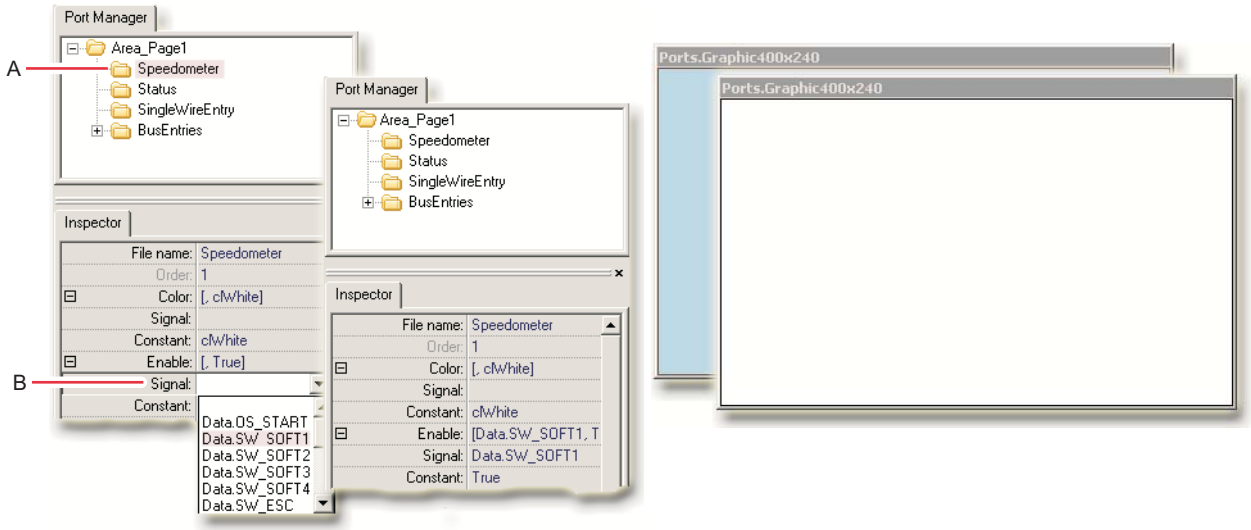


4. **Constant**（定数）のスクリーン・エリアプロパティを変更します。

   A. **Port Manager** タブで、プロパティを変更したいスクリーン・エリアをクリックします。

   レイアウトペインに選択したスクリーン・エリアが表示されます。

   B. **Inspector** タブで、プロパティを変更したい **Constant** をクリックします。表示されたリストから、新しいプロパティを選択します。



5. **Signal**（信号）で画面のプロパティを制御します。

   A. **Port Manager** タブで、プロパティを信号で制御したいスクリーン・エリアをクリックします。

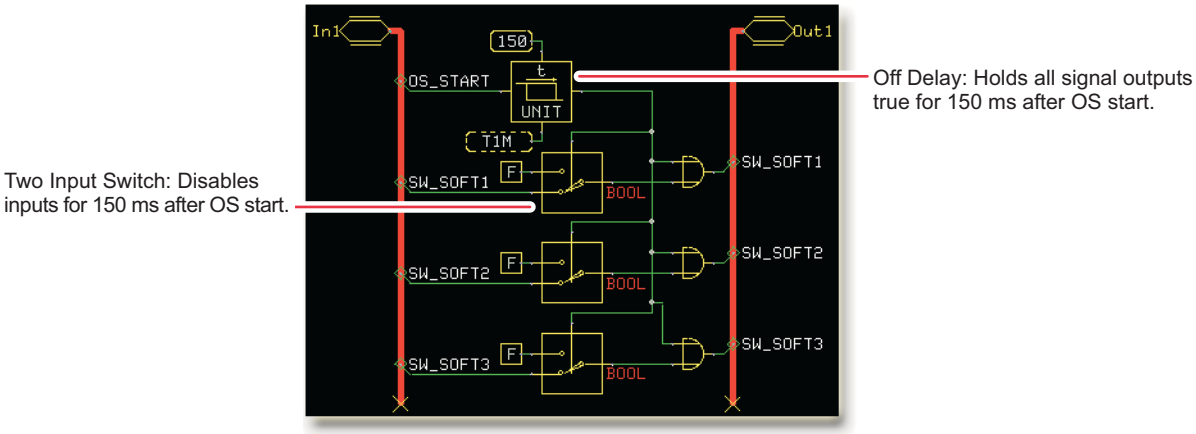
   レイアウトペインに選択したスクリーン・エリアが表示されます。

   B. **Inspector** タブで、制御したいプロパティの **Signal** をクリックします。表示されたリストから、信号を選択します。

Define Areasページ/信号により有効化されたスクリーン･エリアの初期化

ディスプレイの電源を入れると、自動的に信号により有効化されたスクリーン･エリアを表示させることができます。電源オン時、信号により有効化されたファイル（スクリーン･エリア）は有効化信号が真になるまで表示されません。

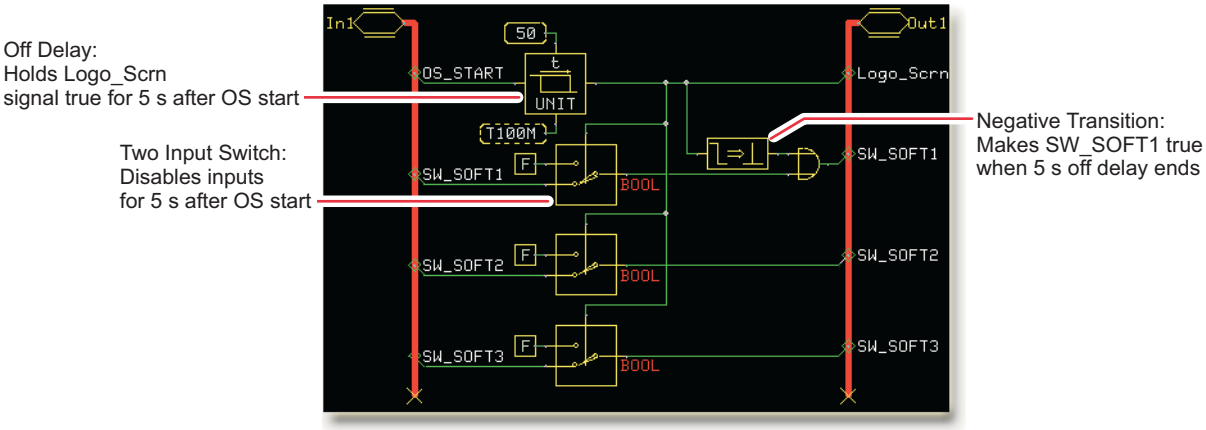
電源オン時の複数スクリーン･エリア表示



ディスプレイの電源を入れると上図ロジックは、

- **SW_SOFT1**、**SW_SOFT2**、**SW_SOFT3** 信号により有効となったスクリーン･エリアを表示します。
- **SW_SOFT1**、**SW_SOFT2**、**SW_SOFT3** 入力を 150 ms 間無効にします。

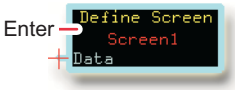
電源オン時の会社ロゴ付きスクリーン･エリア表示



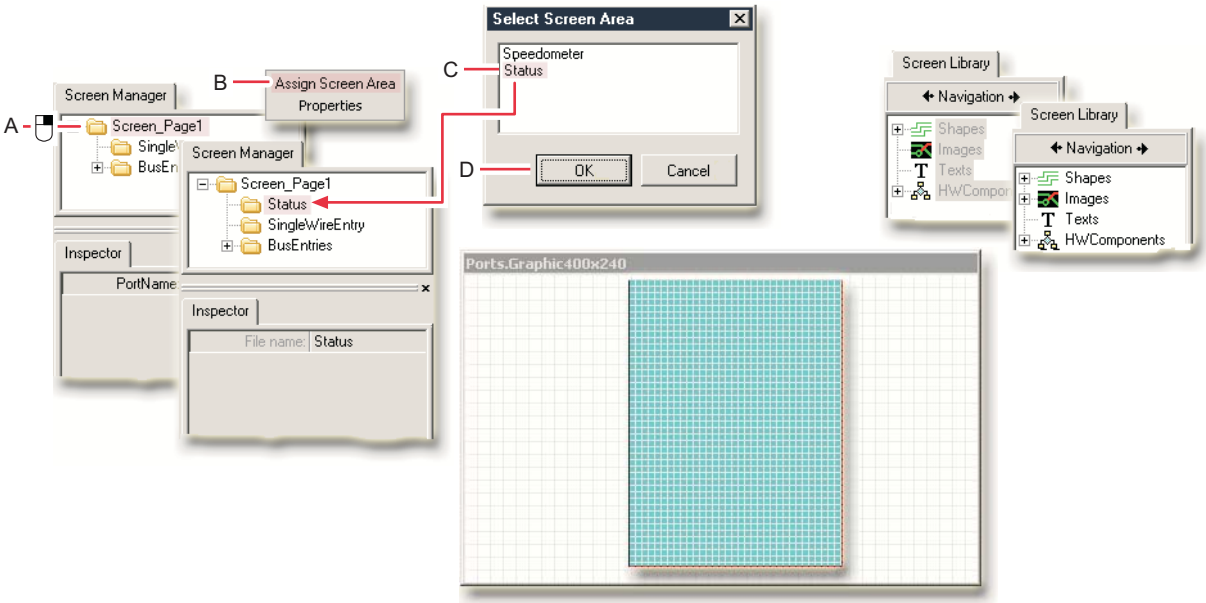
ディスプレイの電源を入れると上図ロジックは、

- **Logo_Scrn** 信号により有効となったスクリーン･エリアを 5 秒間表示します。
- **SW_SOFT1**、**SW_SOFT2**、**SW_SOFT3** 入力を 5 秒間無効にします。
- 5 秒後、**SW_SOFT1** 信号により有効となったスクリーン･エリアを表示します。

Define Screenページ/スクリーン･エリアの割当



1. **Define Screen** ページに入ります。



2. スクリーン･エリアを割当てます。

   A. **Screen Manager** タブで、**Screen_Page1** を右クリックします。

   B. 開いたポップアップメニューで、**Assign Screen Area** をクリックします。

   C. 開いた **Select Screen Area** ウィンドウで、割当てたいスクリーン･エリアをクリックします。

   このウィンドウは **Screen Areas** ページで作成した全スクリーン･エリアを一覧表示します。

   D. **Select Screen Area** ウィンドウで、**OK** をクリックします。

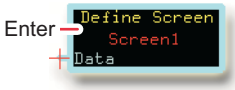
   レイアウトペインは先ほど割当てたスクリーン･エリアを表示します。

   レイアウトペインは **Define Areas** ページで割当てた **HWPortName** の画素寸法に合うようにサイズ変更も行います。

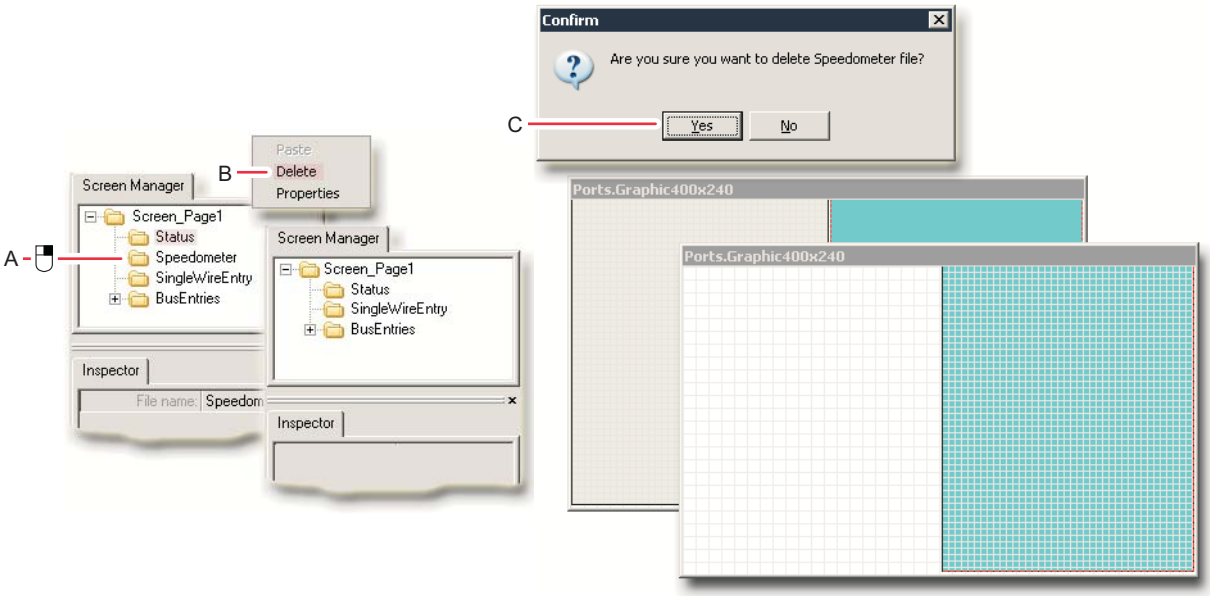
   Screen Library タブのグレーの項目は、最初のスクリーン･エリアを割当てると直ちに使用可能になります。

   さらにスクリーン･エリアを割当てるときは、この手順を繰返します。

Define Screenページ/スクリーン・エリアの削除



1. **Define Screen** ページに入ります。



2. スクリーン・エリアを削除します。

   A. **Screen Manager** タブで、削除したいスクリーン・エリアを右クリックします。

   B. 開いたポップアップメニューで、**Delete** をクリックします。

   C. 開いた **Confirm** ウィンドウで、**Yes** をクリックします。

   スクリーン・エリアがレイアウトペインから消えます。

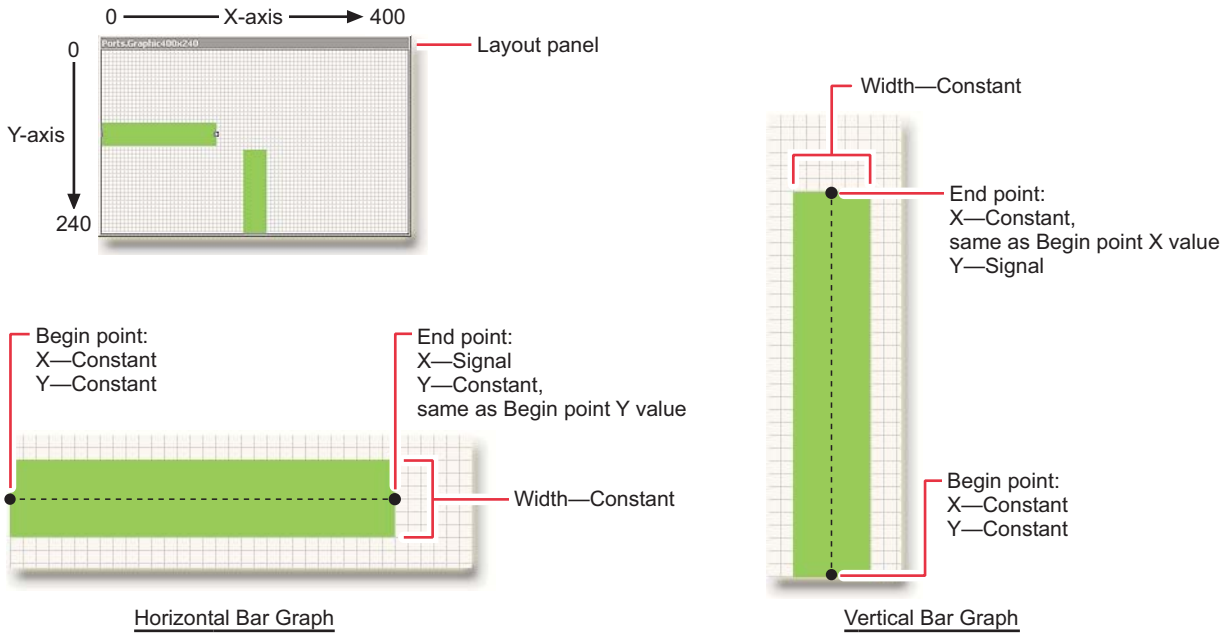
   他のスクリーン・エリアを削除するときは、この手順を繰返します。

Define Screenページ/バーグラフの作成

以下の図と表は、**Line** を用いて横と縦のバーグラフを作成するのに一般的に必要な **Inspector** タブ設定を示したものです。

Screen Library タブの **Shapes** ツリーから **Line** を選択します。

- 横のバーグラフは X 軸方向にサイズが変化します。
- 縦のバーグラフは Y 軸方向にサイズが変化します。



バーグラフの Inspector 基本設定

バーグラフの種類	Begin Point（始点）		End Point（終点）		幅
	X	Y	X	Y	
横	Constant	Constant	Signal	Constant－Y 軸 Begin Point と同じ	Constant
縦	Constant	Constant	Constant－X 軸 Begin Point と同じ	Signal	Constant

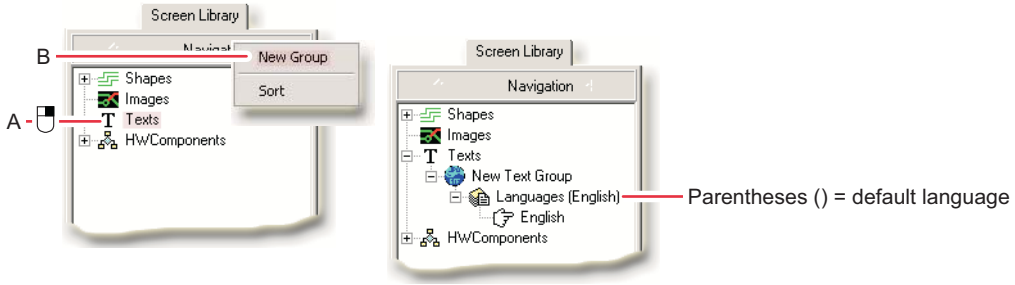
上図に示した横のバーグラフは、

- X 軸 **End point** 信号の画素値が増加すると長くなります。
- X 軸 **End point** 信号の画素値が減少すると短くなります。

上図に示した縦のバーグラフは、

- Y 軸 **End point** 信号の画素値が減少すると高くなります。
- Y 軸 **End point** 信号の画素値が増加すると低くなります。

Define Screenページ/テキストグループの作成



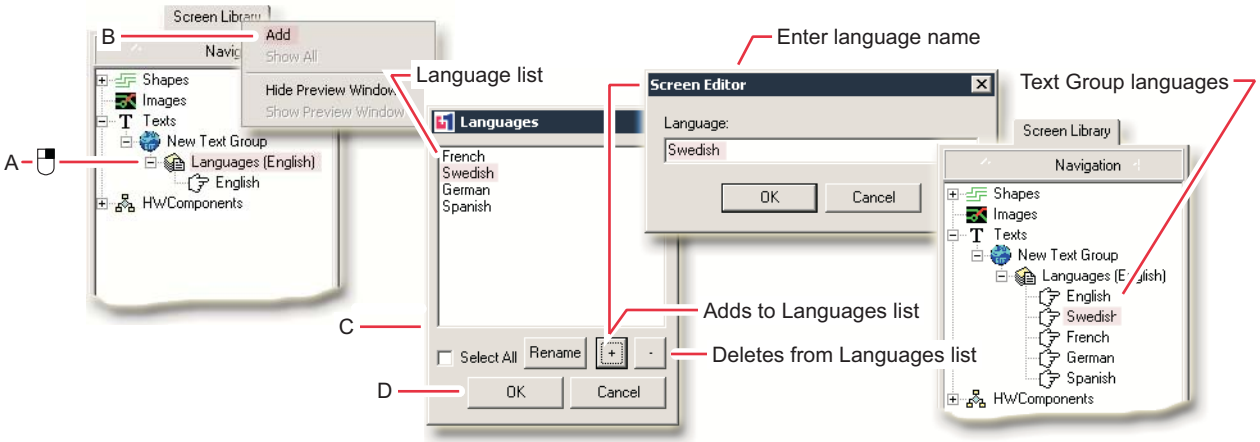
1. **English** を **Text Group** のデフォルト言語に設定します。

   （デフォルト言語はあとで変更できます。詳しくは、***516***ページの「***Define Screen Page/Use the Default Language Setting***」を参照してください。）

   A. **Screen Library** タブで、**Texts** を右クリックします。

   B. 開いたポップアップメニューで、**New Group** をクリックします。

2. 追加言語が不要のときは、ステップ 4 に進んでください。



3. 必要に応じて、**Text Group** に言語を追加します。

   A. **Languages (English)**を右クリックします。

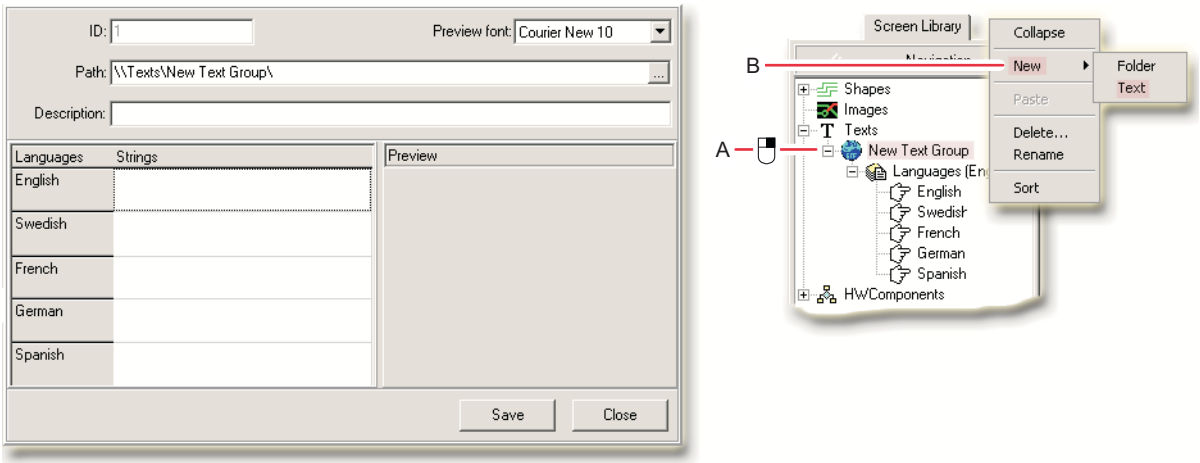
   B. 開いたポップアップメニューで、**Add** をクリックします。

   C. **Languages** ウィンドウを用いて、**Text Group** に言語を追加します。

      – ［**+**］ボタンをクリックすると、**Screen Editor** ウィンドウが開きます。新たな言語名をタイプ入力して、**OK** をクリックします。

      – **Languages** ウィンドウで、追加したい言語をクリックします。

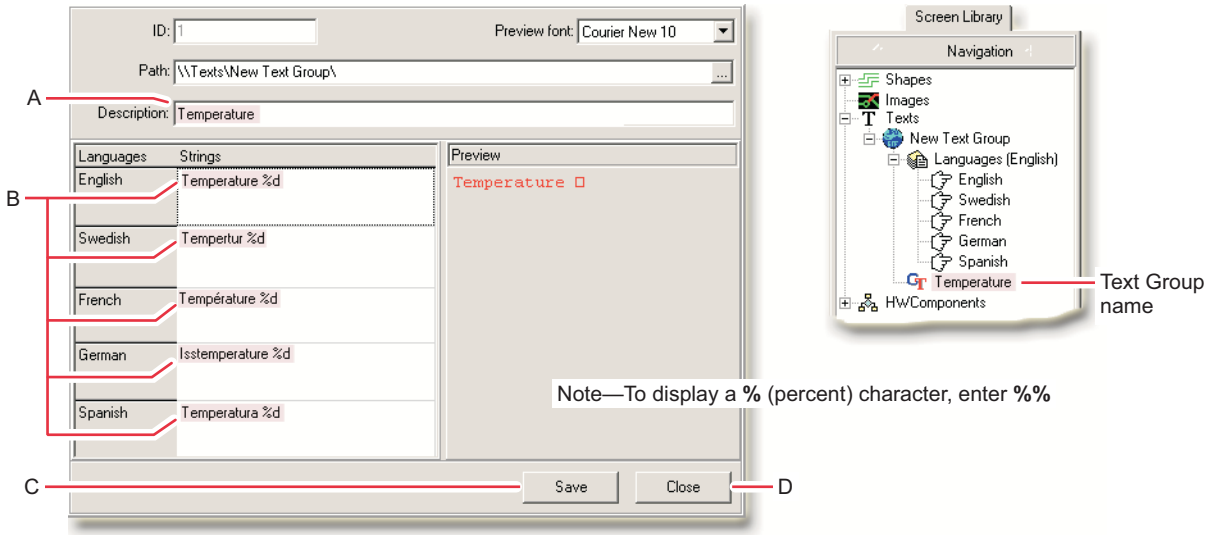
   D. **OK** をクリックして、選択した言語を **Text Group** に追加します。



4. テキスト登録ペインを開きます。

   A. **Screen Library** タブで、**New Text Group** を右クリックします。

   B. 開いたポップアップメニューで、**Text** をクリックしてテキスト登録ペインを表示します。



5. **Text Group** で文字列を定義します。

   A. **Description** に、**Text Group** の参照名を入力します。

   B. **Languages** の各々に文字列を入力します。

   C. **Save** をクリックします。

   Screen Library タブは、**New Text Group** ツリーに **Description** 名を表示します。（必要に応じて、ステップ 4～5 を繰返してさらに **Text Group** を追加します。）

   D. **Close** をクリックして、テキスト登録ペインを閉じます。

Define Screenページ/デフォルト言語設定の使用

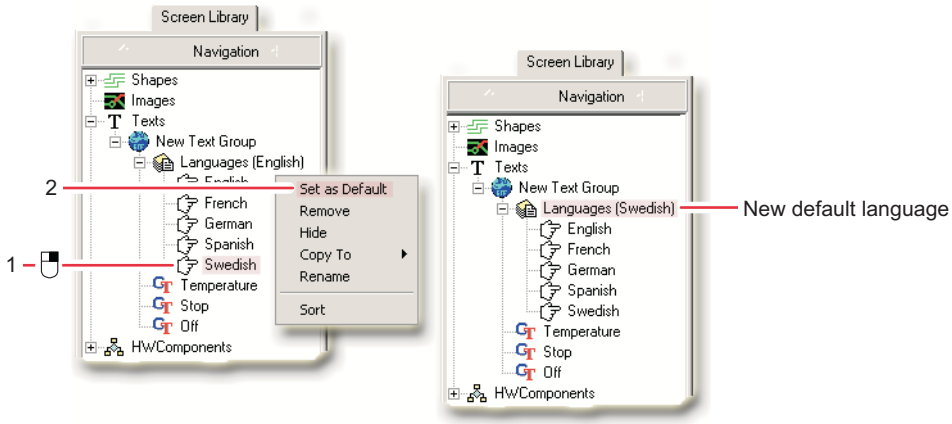
デフォルト言語設定は以下の場合に使用します。

- アプリケーションの **Text Groups** に複数言語による文字列があるとき。
- 言語ごとに別々のダウンロードファイルを作成したいとき。
- エンドユーザーが言語を切換えられないようにしたいとき。

エンドユーザーが言語を切換えられないようにしたいときは、*517*ページの「*Define Screen Page/Switch between Different Languages*」を参照してください。

デフォルト言語の変更方法

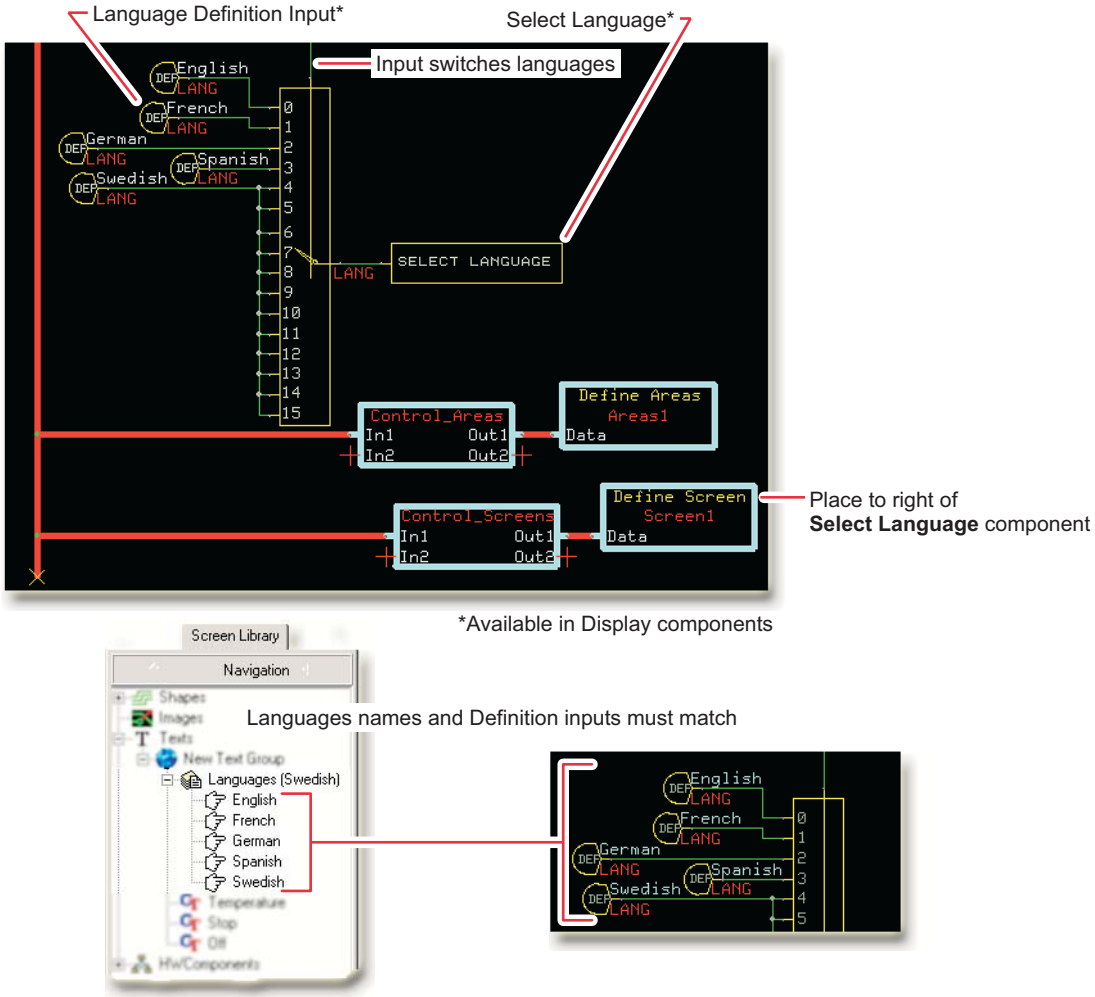
デフォルト言語を変更するときは、以下の手順を実行してください。



1. **Screen Library** タブの **Languages** ツリーで、デフォルト言語にしたい言語を右クリックします。
2. 開いたポップアップメニューで、**Set as Default**（デフォルトに設定）をクリックします。

   Swedish デフォルト言語設定（上図の例）で作成したダウンロードファイルは、スウェーデン語の文字列だけを表示します。

Define Screenページ/言語の切換



以下の場合に、上図を言語切換えロジックの作成の手引きにしてください。

- アプリケーションの **Text Groups** に複数言語による文字列があり、
- エンドユーザーが言語を切換えられるようにしたいとき。

エンドユーザーが言語を切換えられないようにしたいときは、***516***ページの「***Define Screen Page/Use the Default Language Setting***」を参照してください。

Define Screen Page/Import Translated Text into an Application

スクリーンエディタには **Screen Library** タブの **Text Group** からカンマ区切り文字列をインポートまたはエクスポートする機能が備わっています。

インポート/エクスポート機能は多言語に対応したグラフィックディスプレイ用のアプリケーションを作成し、その文字列をある言語から他の言語に翻訳が必要な場合などに有用です。

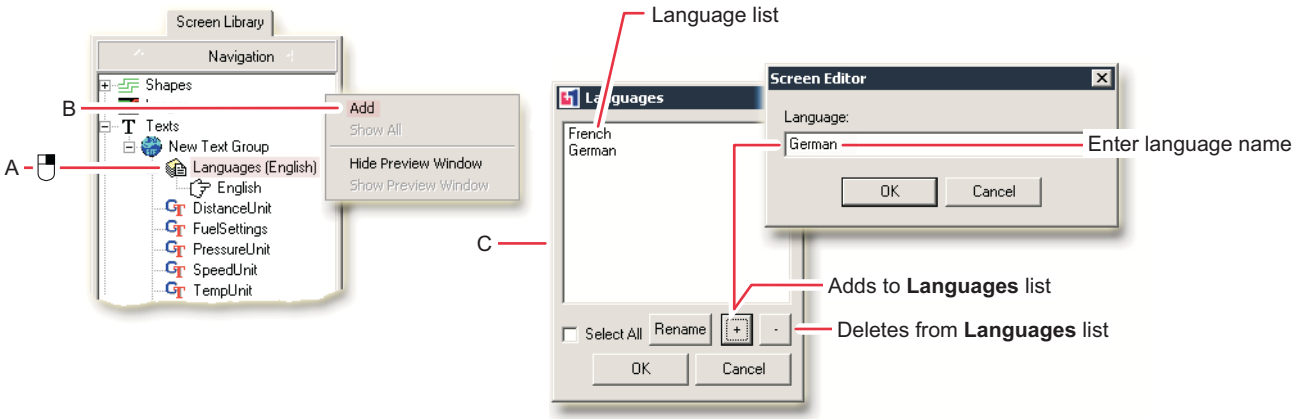
エクスポート/インポート機能を使用すると以下のことができます。

- アプリケーションから、翻訳が必要な文字列を含むカンマ区切り(*.csv)ファイルをエクスポートする
- アプリケーションに、翻訳済みの文字列を含むカンマ区切り文字列をインポートする

この「手引き」では以下のようなグラフィックアプリケーションを想定しています。

- 英語が第一言語
- 英語の文字列をフランス語とドイツ語に翻訳する必要がある

翻訳が必要な文字列を含むCSVファイルのエクスポート



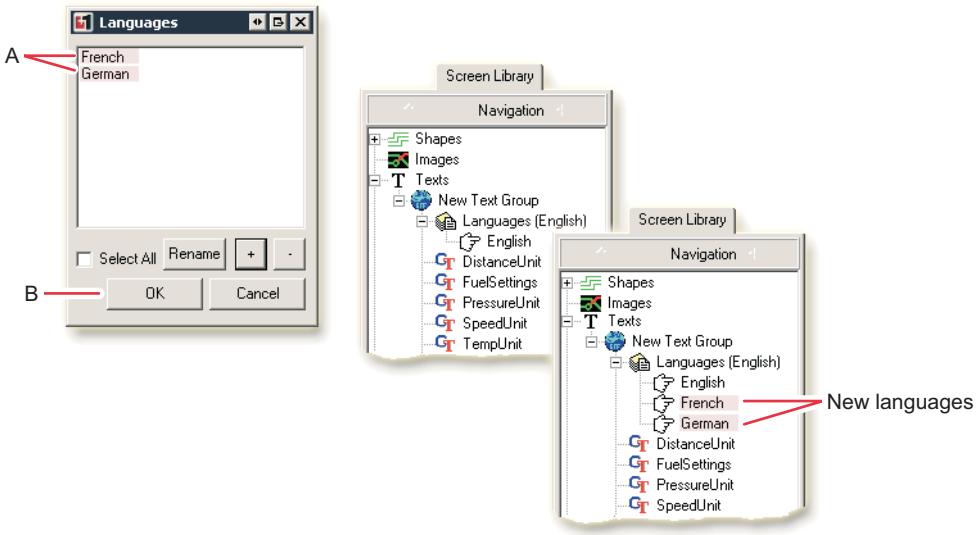
1. **Languages** ウィンドウに翻訳言語を追加します。

   A. **Screen Library** タブの **Text Group** ツリーで、**Languages (English)**.を右クリックします。

   B. ポップアップメニューで **Add** をクリックします。

   C. **Languages** ウィンドウに言語を追加します。

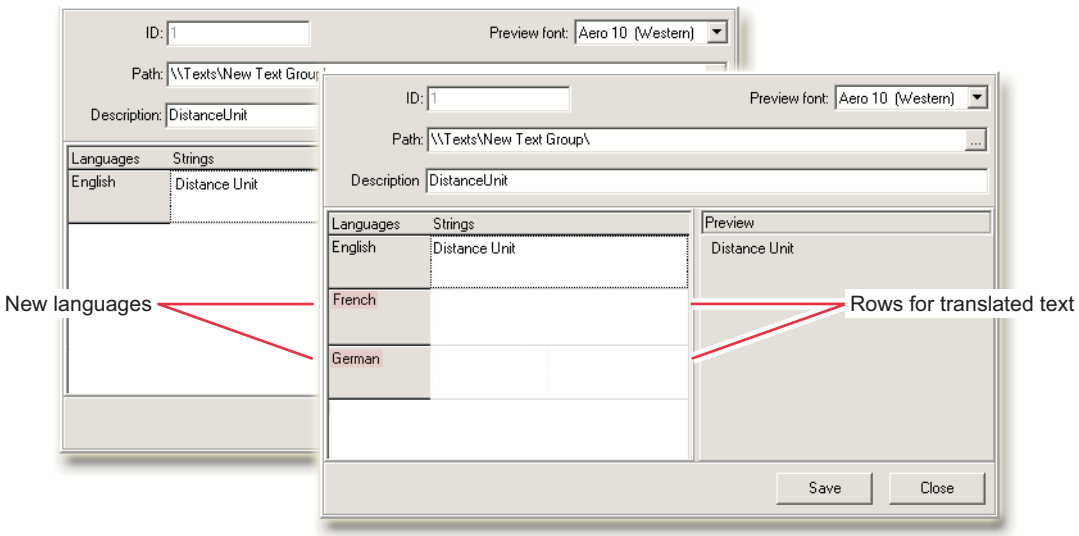
   - ボタンをクリックしてウィンドウを表示します。Click the **+** button to display the **Screen Editor** window.
   - **Screen Editor** ウィンドウで、**Language** フィールド内に言語名をタイプして **OK** をクリックします。
   - **Languages** ウィンドウに全ての必要な言語が追加されるまでこれを繰り返します。



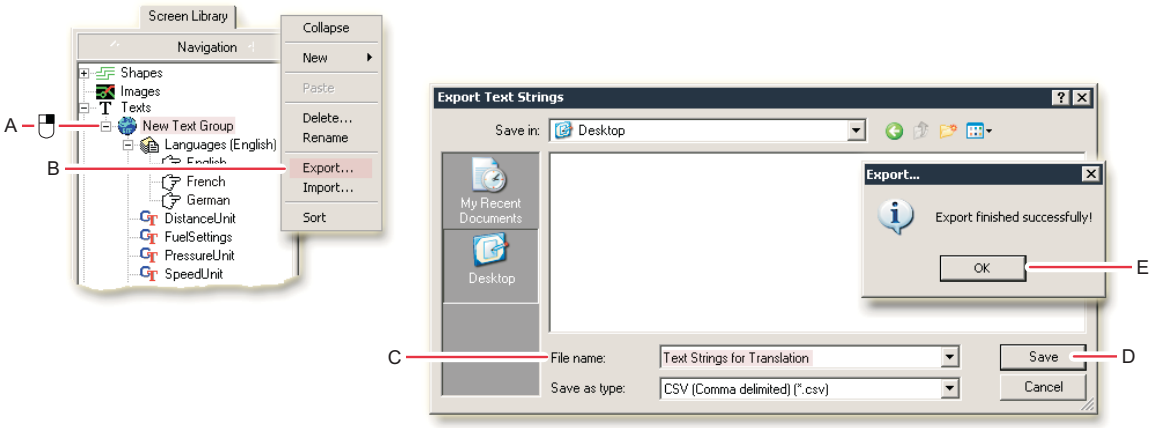
2. **Text Group** ツリーに言語を追加します。

   A. **Languages** ウィンドウで、**Text Group** ツリーに追加したい言語を選択します。

   B. **Languages** ウィンドウで **OK** をクリックして選択した言語を **Text Group** ツリーに追加します。



- 文字列登録ペインの **Languages** の列に先ほど **Text Group** ツリーに追加した新しい言語が表示されています。
- **Strings** の列には翻訳文字列を記入する欄ができています。



3. 翻訳が必要な文字列を含むカンマ区切りテキストファイルをエクスポートします。

   A. **Screen Library** タブで **New Text Group** アイテムを右クリックします。

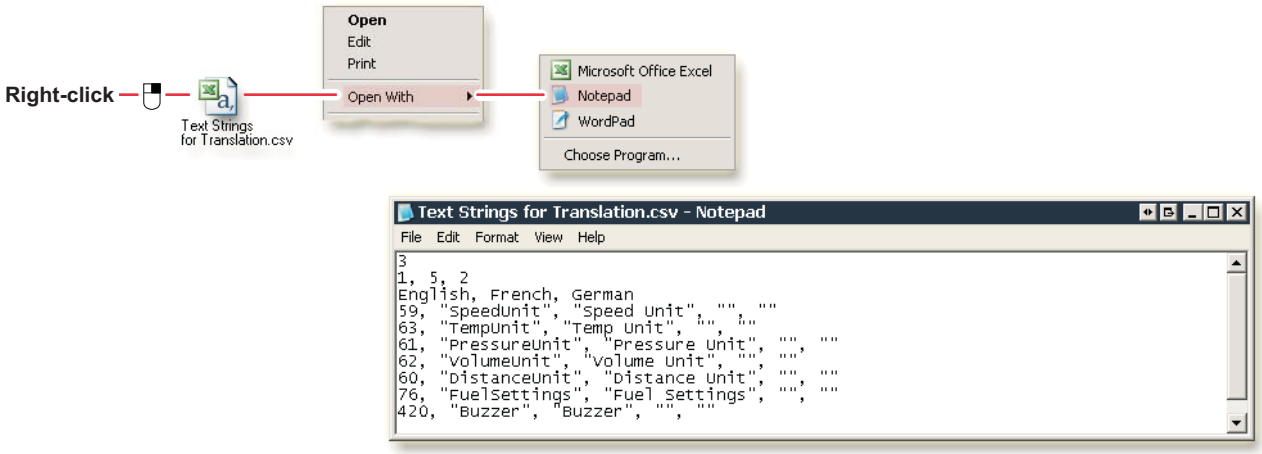
   B. 表示されるポップアップメニューで、**Export** コマンドをクリックして **Export Text Strings** ウィンドウを表示します。

   C. **Export Text Strings** ウィンドウで適切な **File name** を入力します。

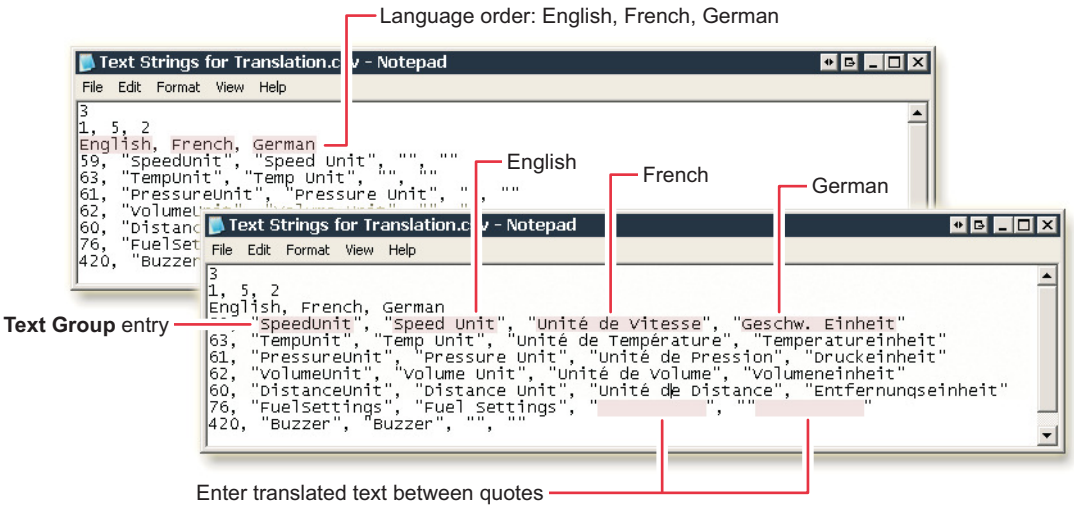
   D.**Export Text Strings** ウィンドウで **Save** をクリックして文字列をエクスポートします。

   E. **OK** をクリックして **Export** ウィンドウを閉じます。

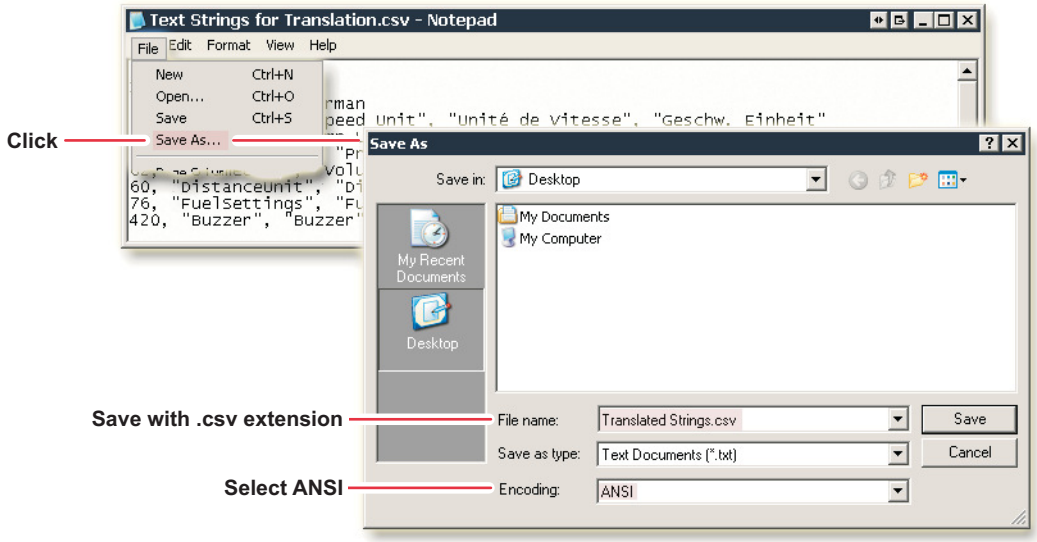
文字列の翻訳



1. **Notepad**（メモ帳）プログラムを用いて翻訳が必要な文字列を含むファイルを開きます。

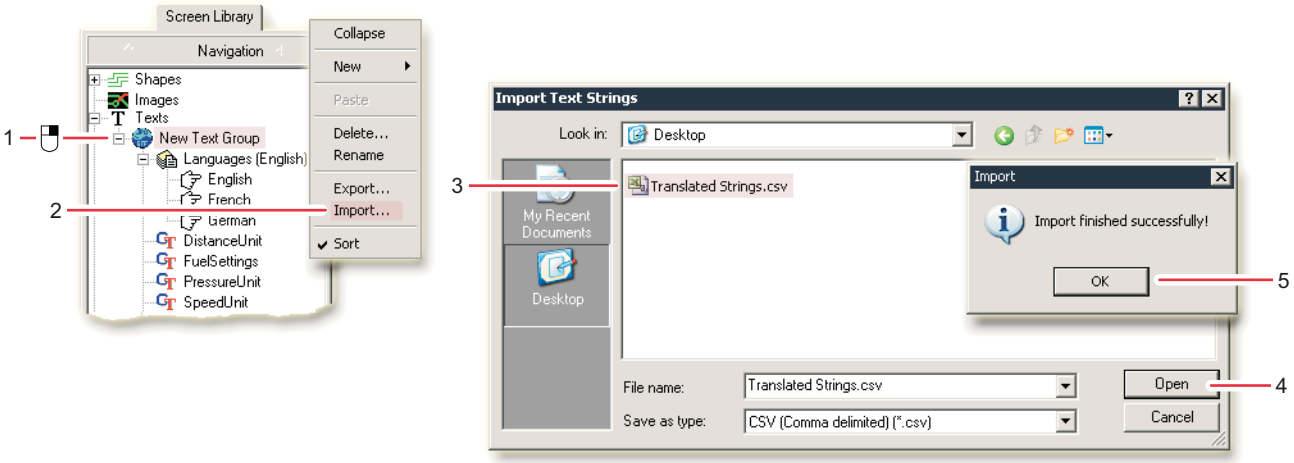


2. 翻訳文字列をテキストファイルに挿入します。
   - 引用符の間に翻訳文字列を入力します。
   - 翻訳文字列を入力するときは上の図で確認した言語の順序に従ってください。



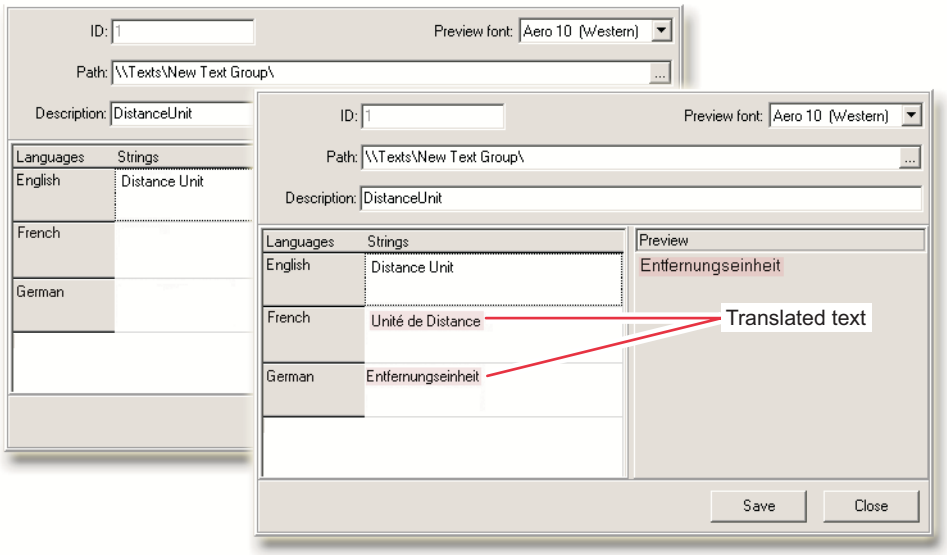
3. 翻訳文字列を保存します。
   - 翻訳文字列をカンマ区切りテキストで保存します。
   - **Encoding** を **ANSI** に選択してファイルを保存します。

翻訳文字列を含むCSVファイルのインポート



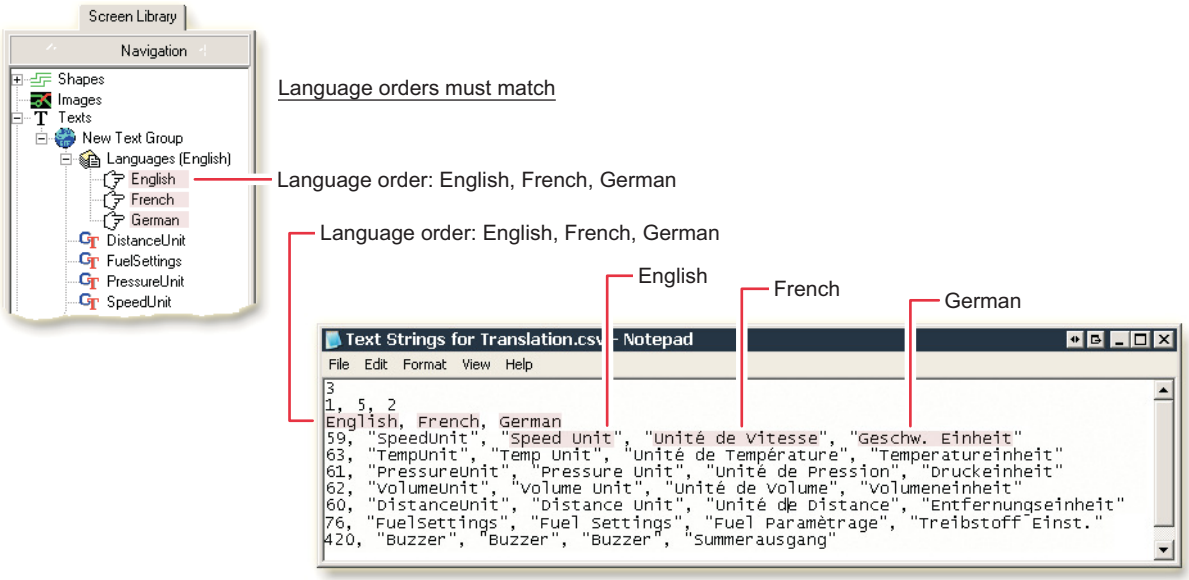
翻訳文字列を含むカンマ区切りテキストファイルをインポートします。

1. **Screen Library** タブで、**New Text Group** アイテムを右クリックします。
2. 表示されたポップアップメニューで、**Import** コマンドをクリックして **Import Text Strings** ウィンドウを表示します。
3. **Import Text Strings** ウィンドウで、翻訳文字列を含む CSV ファイルをクリックします。
4. **Import Text Strings** ウィンドウで **Open** をクリックして文字列をインポートします。
5. **OK** をクリックして **Import** を閉じます。

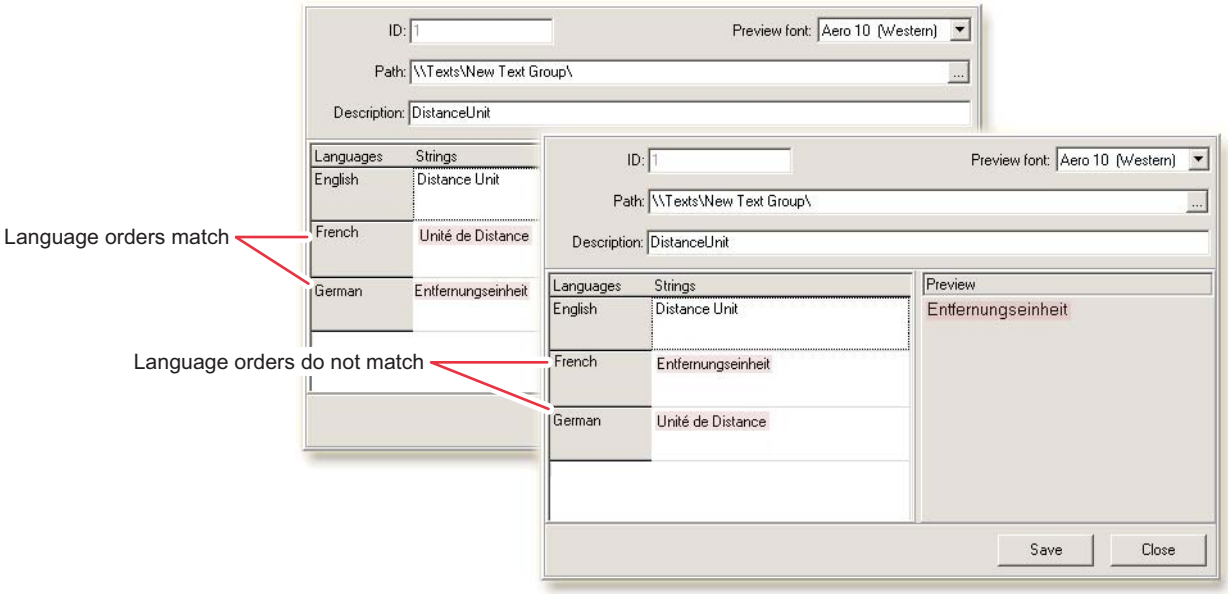


文字列登録ペインの **Strings** 列に翻訳文字列が表示されています。

言語の順序について



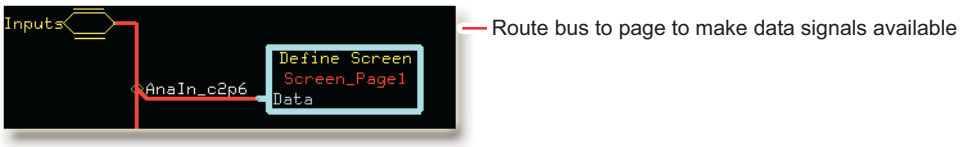
翻訳した文字列を正しくインポートするために **Screen Library** タブ内の言語の順序はカンマ区切りファイル内の言語の順序と一致している必要があります。



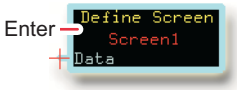
言語の順序が一致しない場合、グラフィックディスプレイは間違った言語を表示します。例えば、ドイツ語を選択するとディスプレイはフランス語の文字列を表示するなどです。

必要に応じ、***520***ページのステップ***2***を繰り返して**Screen Library**タブの言語の順序がカンマ区切りテキストファイル内の言語の順序に一致するようにします。

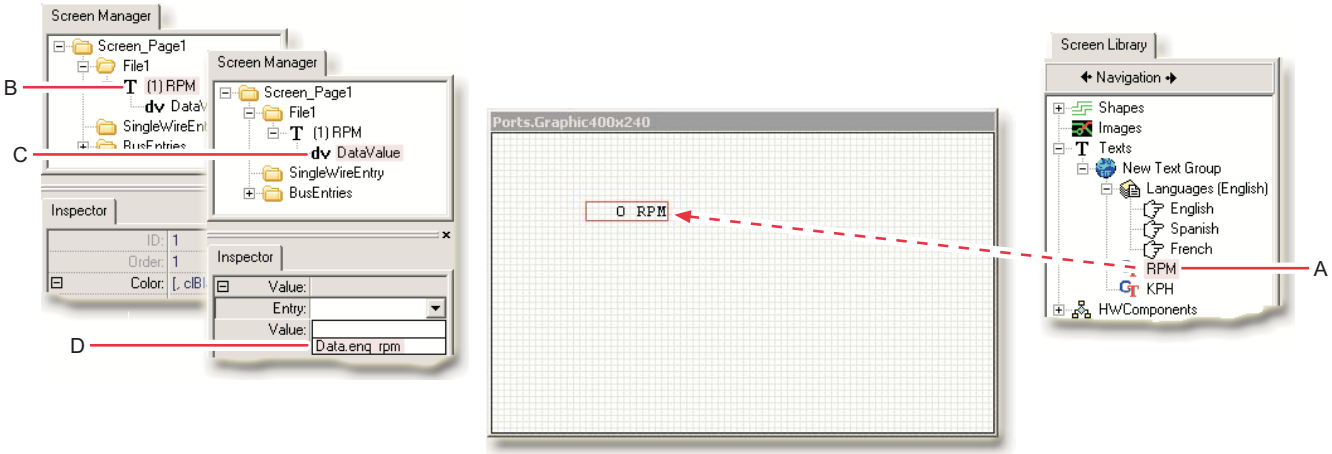
Define Screenページ/データ値の表示



1. 必要なデータ信号のバスを **Define Screen** ページに引き込みます。



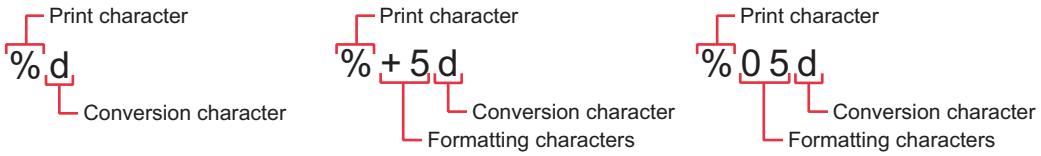
2. **Define Screen** ページに入ります。
3. テキスト登録ペインを使用して、%指定子つきの文字列を作成して登録します。（***514***ページの「***Define Screen Page/Create a Text Group***」を参照してください。）



4. 表示したいデータ値を選択します。
   A. 登録したテキストを **Screen Library** タブからレイアウトペインにドラッグします。
   B. **Screen Manager** タブで、新たな **T**（テキスト）入力をクリックして内容を表示します。
   C. **Screen Manager** タブで、**dv DataValue** をクリックします。
   D. **Inspector** タブで、表示したい値の信号をクリックします。

Define Screenページ/データ値のフォーマット

データ値表示に使用するエレメントを下図に示します。



- データ値には以下が必要です。
  - %指定子(Print character)
  - 表示したいデータ値に合った変換指定子(Conversion character)
- 必要に応じてフォーマット指定子(Formatting character)を使用して、データの見た目を整えます。

変換指定子のプロパティ

変換指定子	表示	データ型						最小値	最大値
c	Char (単一キャラクタ)	U8	—	—	—	—	—	—	—
d or i	Integer, 10 進表記	U8	S8	—	S16	—	—	−32768	32767
ld or li	Long integer, 10 進表記	U8	S8	U16	S16	—	S32	−2147483648	2147483647
u	Unsigned integer, 10 進表記	U8	—	U16	—	—	—	0	65535
lu	Unsigned long integer, 10 進表記	U8	—	U16	—	U32	—	0	4294967295
x or X	Unsigned integer, 16 進表記	U8	—	U16	—	—	—	—	—
lx or lX	Unsigned long integer, 16 進表記	U8	—	U16	—	U32	—	—	—
o	Unsigned integer, 8 進表記	U8	—	U16	—	—	—	—	—
lo	Unsigned long integer, 8 進表記	U8	—	U16	—	U32	—	—	—

特殊フォーマット指定子

文字	説明	フォーマッティング例
スペース	正の表示値の前に空白を、負の表示値の前にー記号を入れます。	項目 2
+	正の表示値の前に＋記号を、負の表示値の前にー記号を入れます。	項目 3
n	表示値を右につめます。n は値の前の最大スペース数です。値が大きくなると、スペースは減少します。	項目 4
-***n***	表示値を左につめます。n は値の後の最大スペース数です。値が大きくなると、スペースは減少します。	項目 5
0***n***	表示値の左につけるゼロの最大数を設定します。n はゼロの数です。	項目 6
#	代替表示値 #o：8 進数出力で、値の前に「0」がつきます。 #X：　大文字の 16 進数出力で、値の前に 0X がつきます。	——

*フォーマッティング例（**S16** インプット）*

項目	フォーマッティング	入力値	表示値	入力値	表示値	入力値	表示値
1	%␣d	0	0	111	111	-111	−111
2	%+d	0	+0	111	+111	-111	−111
3	%5d	0	0	111	111	-111	−111
4	%-5d	0	0	111	111	-111	−111
5	%05d	0	00000	111	00111	-111	−0111

Define Screenページ/画像リストの表示

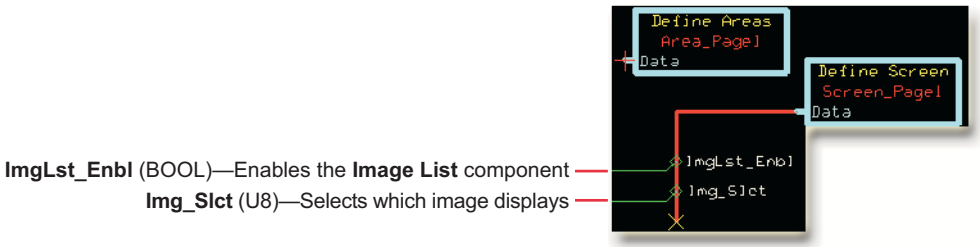
この手順の前に以下の割当がされている必要があります。

- ハードウェアポートの **Define Areas** ブロックへの割当。

  詳しくは、*504*ページの「*Define Areas Page/Assign a Hardware Port*」を参照してください。

- スクリーン・エリアの **Define Screen** ブロックへの割当。

  詳しくは、*511*ページの「*Define Screen Page/Assign a Screen Area*」を参照してください。



1. **Image List** コンポーネントの制御ロジックを作成します。

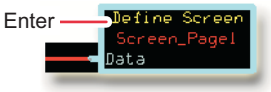
   通常、このロジックは以下を含みます。

   - **Image List** コンポーネントの **Enable** プロパティを制御するブール信号。

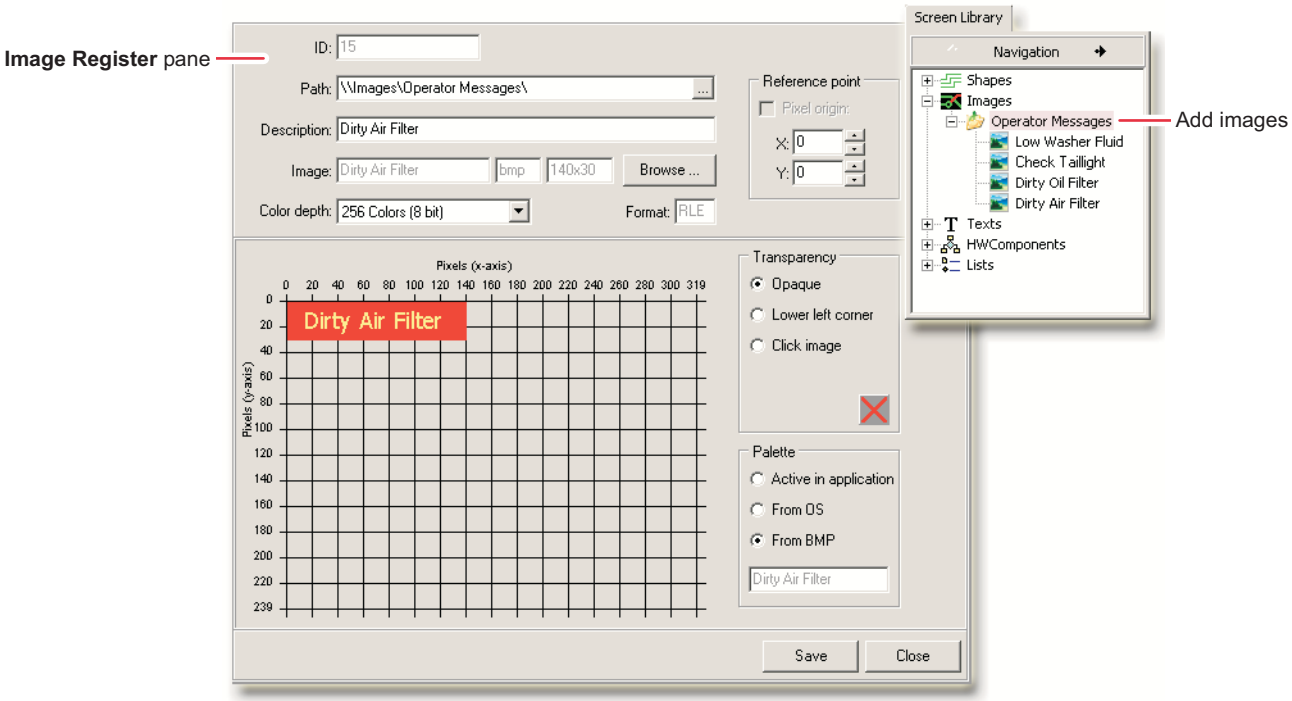
     この例では、ブール信号 **ImgLst_Enbl**（Image List Enable）が T（True）のとき、**Image List** コンポーネントを表示します。

   - **Image List** コンポーネントの **ActiveIndex** プロパティを制御する整数信号。

     この例では、U8 **Img_Slct**（Image Select）信号の値が **Image List** コンポーネントの表示する画像を制御します。

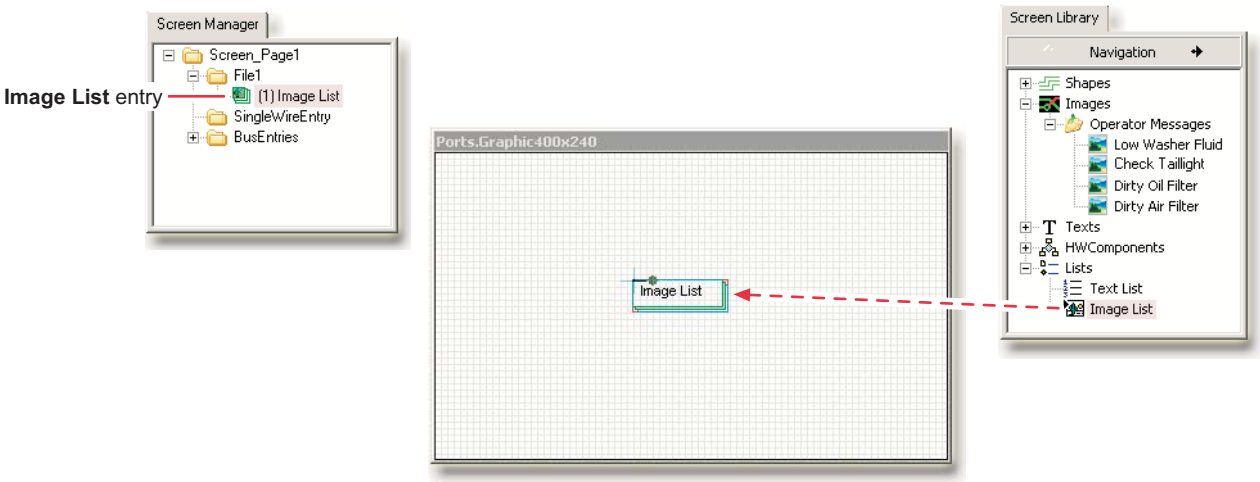


2. **Define Screen** ブロックに入ります。

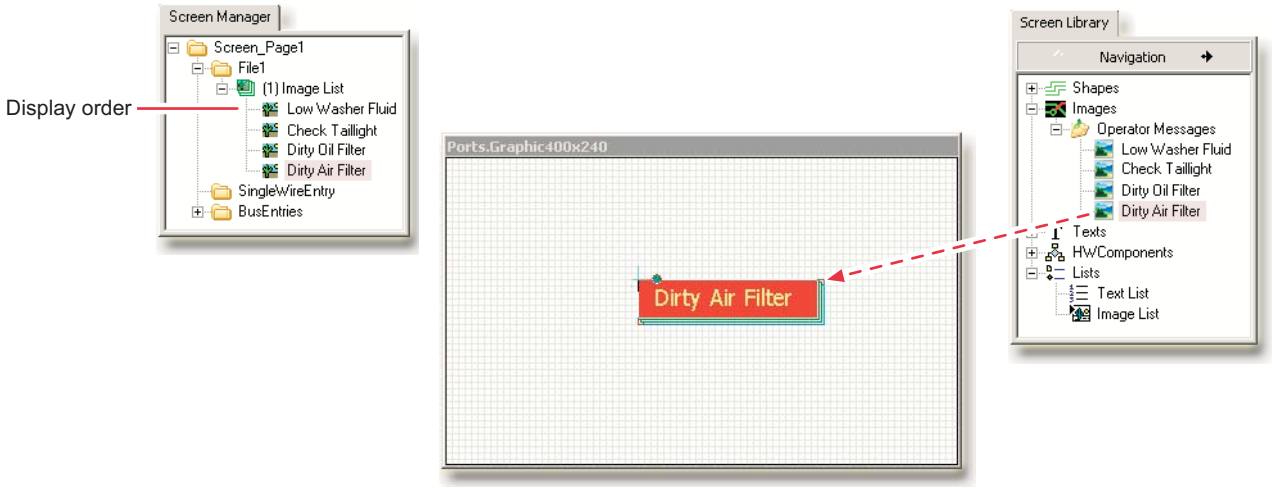


3. **Image Register** ペインを使用して、**Screen Library** タブの **Image** ツリーに画像を追加します。

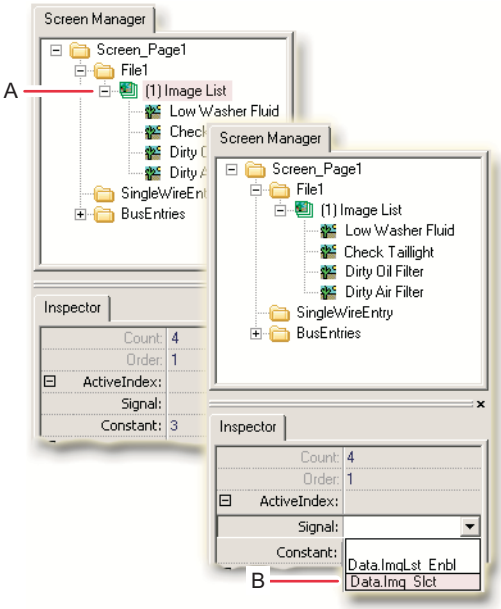
   Image Register ペインの詳細については、*391* ページの「*Define Screen ページ / Image Register*」を参照してください。



4. **Image List** コンポーネントを **Screen Library** タブからレイアウトペインにドラッグします。
   - 画像を表示したい場所に **Image List** コンポーネントを配置します。
   - **Screen Manager** タブに **Image List** が表示されます。



5. **Screen Library** タブから画像を **Image List** コンポーネントにドラッグします。
   - **Image List** コンポーネントに加えた画像のそれぞれに、**Image List** コンポーネントの階層状の境界線がつきます。
   - **Image List** コンポーネントに加えた画像の名前が **Screen Manager** タブの **Image List** ツリーに表示されます。



6. **Image List** コンポーネントの表示画像を制御する **ActiveIndex** 信号を選択します。

   A. **Screen Manager** タブで、**Image List** をクリックします。

   B. **Inspector** タブで、**Image List** コンポーネントの **ActiveIndex** プロパティを制御する信号を選択します。

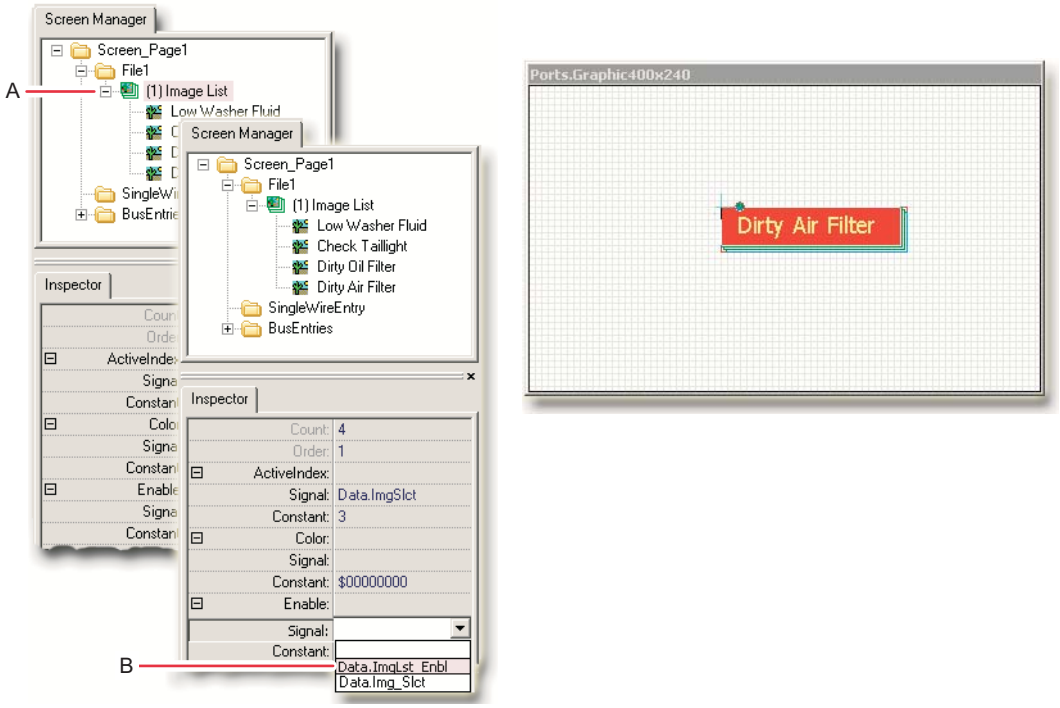
   Image List コンポーネントに画像を追加した順序で、各画像の初期 **Order**（順序）値が設定されます。

   （**Image List** ツリーの画像を移動して、画像の **Order** を変更することができます。）

   画像の **Order** 値と **ActiveIndex** 値が等しいときに、画像が表示されます。

   この例では、**Data.Img_Slct** 信号が **ActiveIndex** プロパティを設定します。

   - **Low Washer Fluid** 画像の **Order** 値は 0 です。**Data.Img_Slct** 信号の値が 0 のとき、この画像を表示します。
   - **Check Taillight** 画像の **Order** 値は 1 です。**Data.Img_Slct** 信号の値が 1 のとき、この画像を表示します。
   - **Dirty Oil Filter** 画像の **Order** 値は 2 です。**Data.Img_Slct** 信号の値が 2 のとき、この画像を表示します。
   - **Dirty Air Filter** 画像の **Order** 値は 3 です。**Data.Img_Slct** 信号の値が 3 のとき、この画像を表示します。



7. **Image List** コンポーネントの表示を有効にする **Enable** 信号を選択します。

   A. **Screen Manager** タブで、**Image List** をクリックします。

   B. **Inspector** タブで、**Enable** プロパティを設定する信号を選択します。

   Enable プロパティが T のときに、**Text List** コンポーネントが表示されます。

   この例では、

   – **Data.ImgList_Enbl** 信号が **Enable** プロパティを設定します。

   – **Data.ImgList_Enbl** 信号が T でない限り、**Image List** コンポーネントの画像は一切表示しません。

   Image List コンポーネントの常時表示を有効にするには、その **Enable** プロパティを **Constant** 値 **T** に設定します。

8. アプリケーションをコンパイルして、ディスプレイにダウンロードします。

Define Screenページ/テキストリストの表示

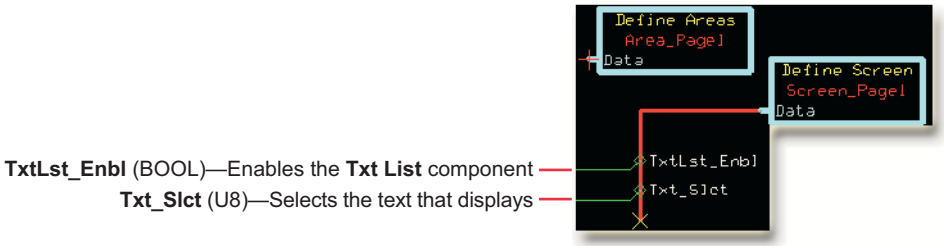
この手順の前に以下の割当がされている必要があります。

- ハードウェアポートの **Define Areas** ブロックへの割当。

  詳しくは、*504*ページの「*Define Areas Page/Assign a Hardware Port*」を参照してください。

- スクリーン･エリアの **Define Screen** ブロックへの割当。

  詳しくは、*511*ページの「*Define Screen Page/Assign a Screen Area*」を参照してください。



1. **Text List** コンポーネントの制御ロジックを作成します。

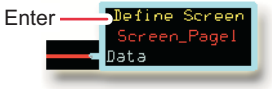
   通常、このロジックは以下を含みます。

   - **Text List** コンポーネントの **Enable** プロパティを制御するブール信号。

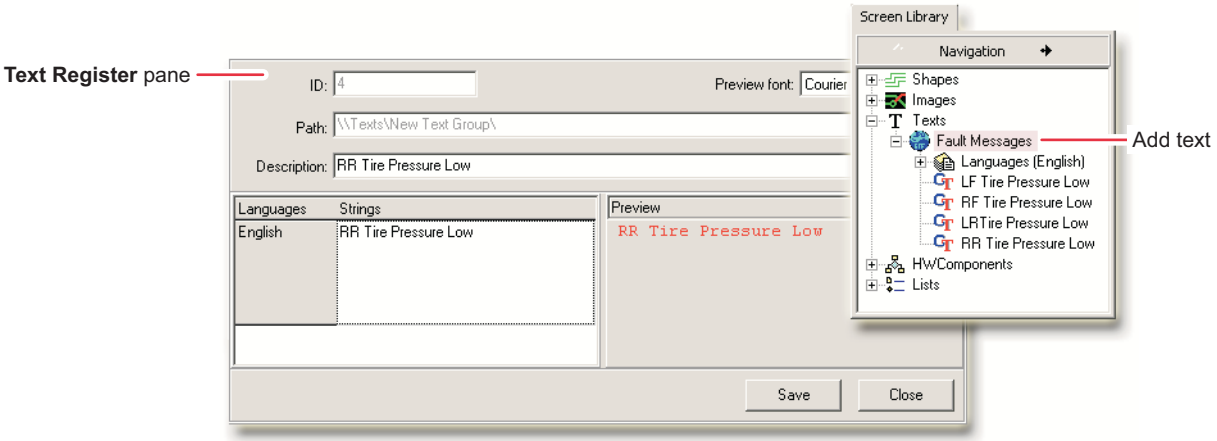
     この例では、ブール信号 **TxtLst_Enbl**（Text List Enable）が T のとき、**Text List** コンポーネントを表示します。

   - **Text List** コンポーネントの **ActiveIndex** プロパティを制御する整数信号。

     この例では、U8 **Txt_Slct**（Text Select）信号の値によって **Text List** コンポーネントの表示する文字列を制御します。

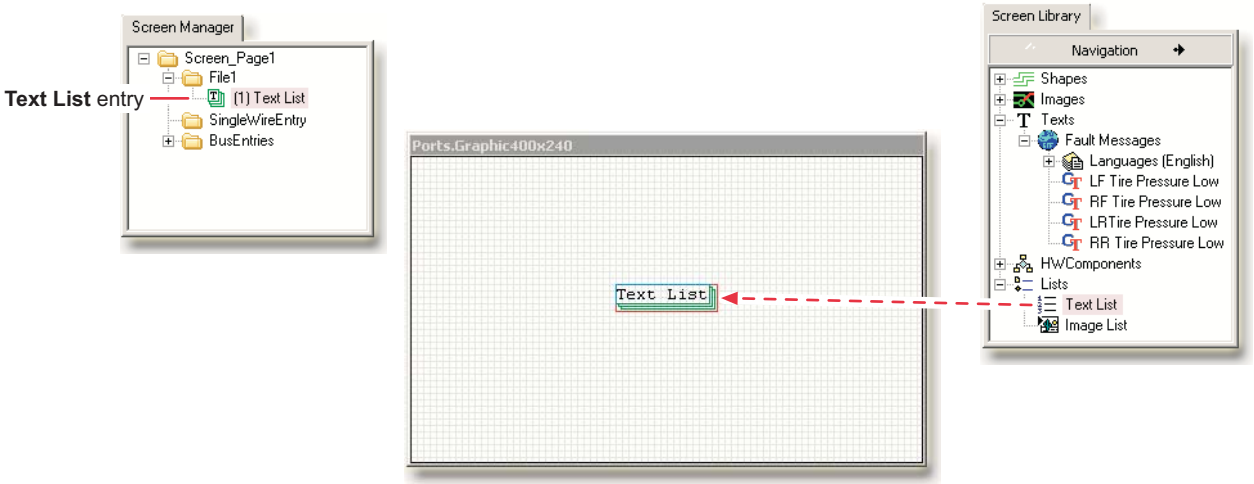


2. **Define Screen** ブロックに入ります。

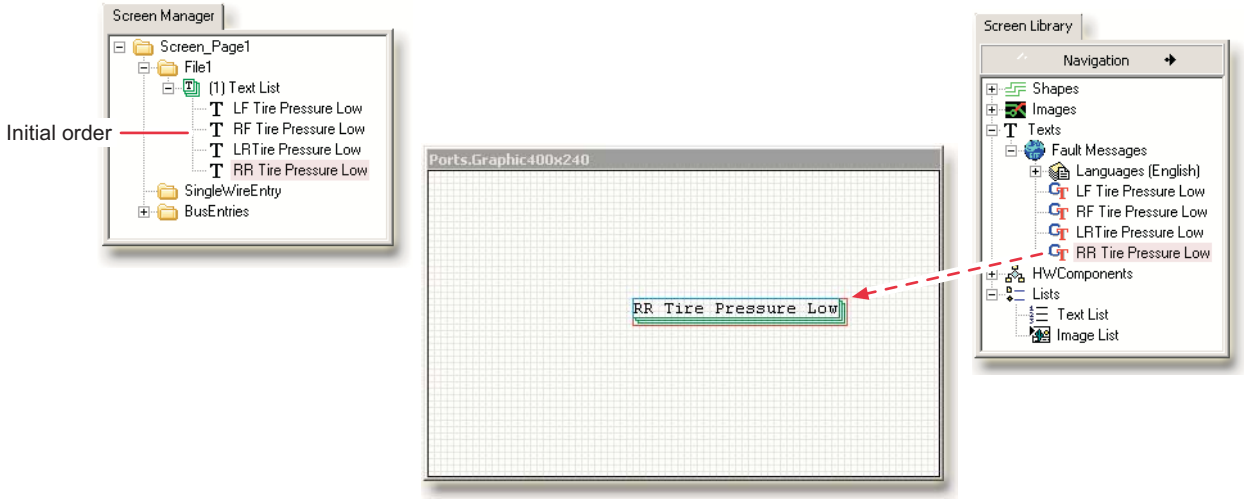


3. **Text Register** ペインを使用して、**Screen Library** タブの **Texts** ツリーにテキストを追加します。

   Text Register ペインの詳細については、*394* ページの「*Define Screen ページ/テキスト登録*」を参照してください。

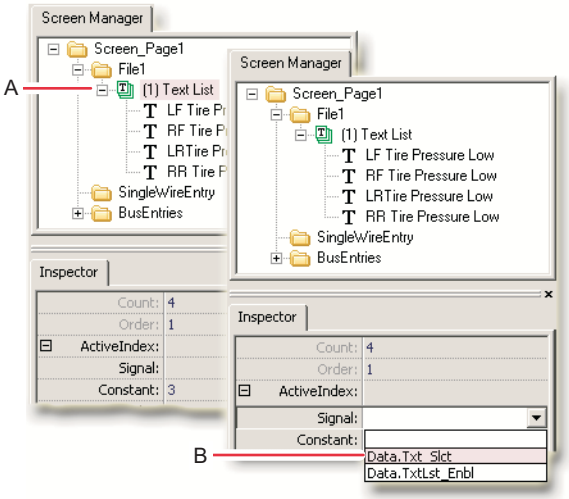


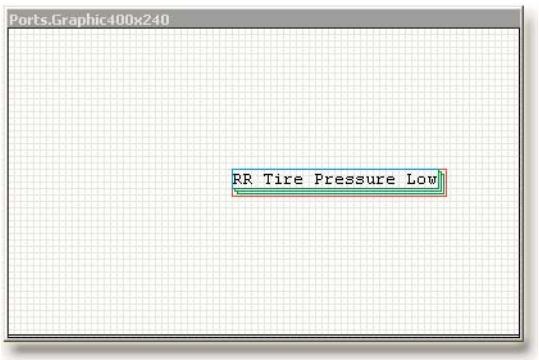
4. **Text List** コンポーネントを **Screen Library** タブからレイアウトペインにドラッグします。
   - テキストを表示したい場所に **Text List** コンポーネントを配置します。
   - **Screen Manager** タブに **Text List** が表示されます。



5. **Screen Library** タブからテキストを **Text List** コンポーネントにドラッグします。

   - **Text List** コンポーネントに加えたテキストのそれぞれに、**Text List** コンポーネントの階層状の境界線がつきます。
   - **Text List** コンポーネントに加えたテキストの名前が **Screen Manager** タブの **Text List** ツリーに表示されます。





6. **Text List** コンポーネントの表示テキストを制御する **ActiveIndex** 信号を選択します。

   A. **Screen Manager** タブで、**Text List** をクリックします。

   B. **Inspector** タブで、**Text List** コンポーネントの **ActiveIndex** プロパティを制御する信号を選択します。

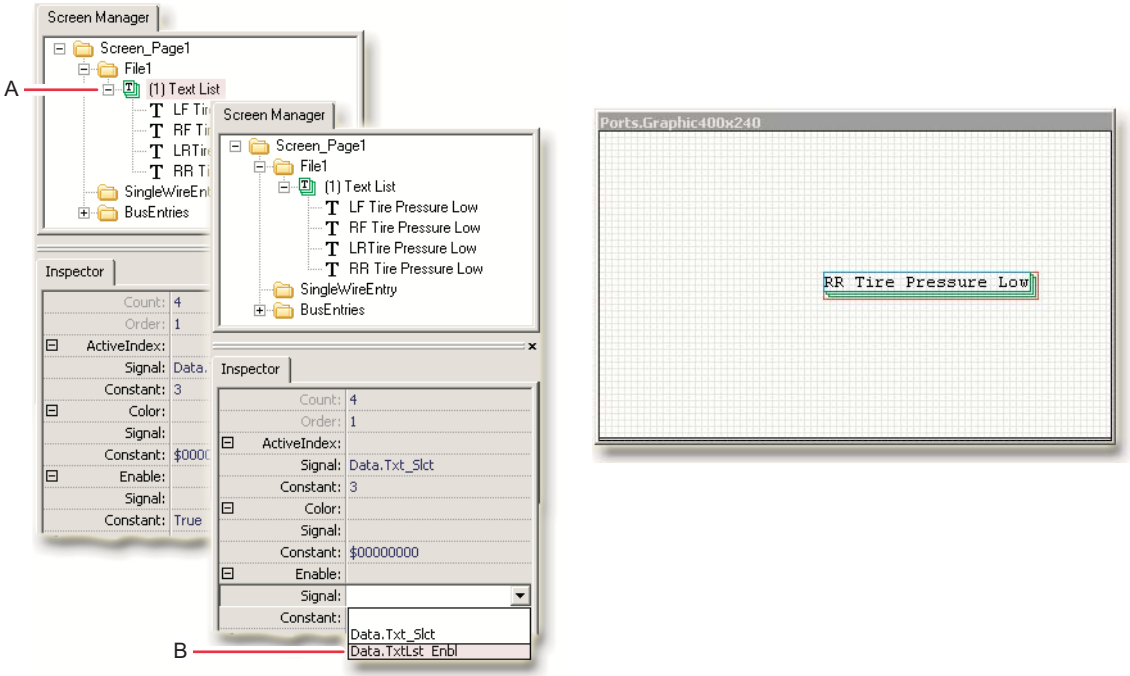
   Text List コンポーネントにテキストを追加した順序で、各テキストの初期 **Order**（順序）値が設定されます。

   （**Text List** エントリのテキストを移動して、テキストの **Order** を変更することができます。）

   テキストの **Order** 値と **ActiveIndex** 値が等しいときに、テキストが表示されます。

   この例では、**Data.Txt_Slct** 信号が **ActiveIndex** プロパティを設定します。

   - **LF Tire Pressure Low** の **Order** 値は 0 です。**Data.Txt_Slct** 信号の値が 0 のとき、このテキストを表示します。
   - **RF Tire Pressure Low** の **Order** 値は 1 です。**Data.Txt_Slct** 信号の値が 1 のとき、このテキストを表示します。
   - **LR Tire Pressure Low** の **Order** 値は 2 です。**Data.Txt_Slct** 信号の値が 2 のとき、このテキストを表示します。
   - **RR Tire Pressure Low** の **Order** 値は 3 です。**Data.Txt_Slct** 信号の値が 3 のとき、このテキストを表示します。



7. **Text List** コンポーネントの表示を有効にする **Enable** 信号を選択します。

   A. **Screen Manager** タブで、**Text List** をクリックします。

   B. **Inspector** タブで、**Text List** コンポーネントの **Enable** プロパティを制御する信号を選択します。

   Enable プロパティが T のときに、**Text List** コンポーネントが表示されます。

   この例では、

   – **Data.TxtList_Enbl** 信号が **Text List** コンポーネントの **Enable** プロパティを設定します。

   – **Data.TxtList_Enbl** 信号が T でない限り、**Text List** コンポーネントのテキストは一切表示しません。

   Text List コンポーネントの常時表示を有効にするには、その **Enable** プロパティを **Constant** 値 **T** に設定します。

8. アプリケーションをコンパイルして、ディスプレイにダウンロードします。

Define Screenページ/アプリケーションフォントのインストール

スクリーンエディタープログラムには 2 種類の標準システムフォントが含まれています。この「How-To」ではプロジェクトで追加フォントを使えるようにする方法を記述しています。

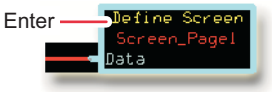
この手順の前に以下の割当がされている必要があります。

- ハードウェアポートの **Define Areas** ブロックへの割当。

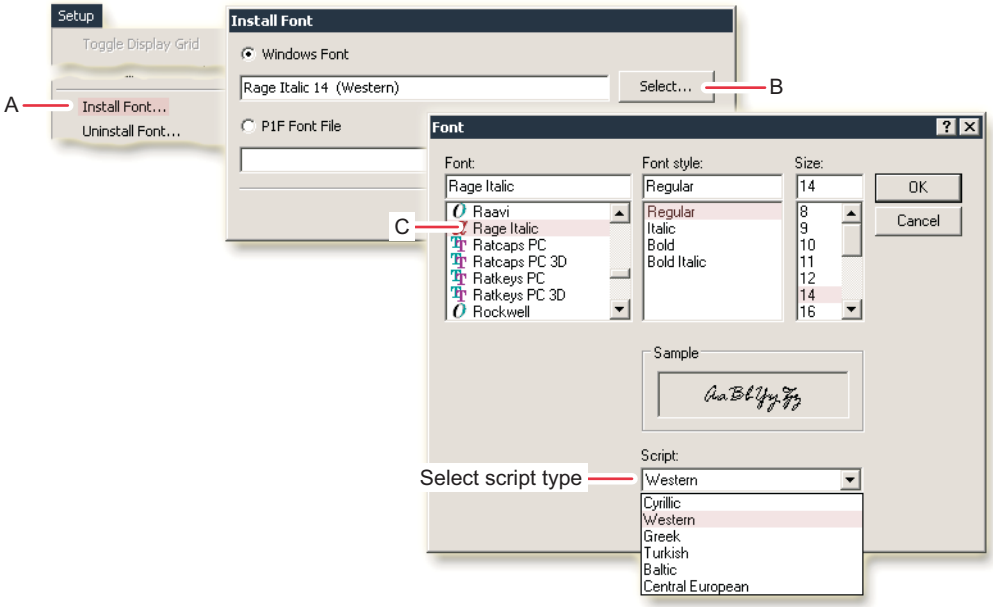
  詳しくは、*504*ページの「*Define Areas Page/Assign a Hardware Port*」を参照してください。

- スクリーン・エリアの **Define Screen** ブロックへの割当。

  詳しくは、*511*ページの「*Define Screen Page/Assign a Screen Area*」を参照してください。



1. **Define Screen** ブロックに入ります。



2. PLUS+1 GUIDE プログラムで追加フォントを使用できるようにします。

   このステップでは、PLUS+1 GUIDE 用 **Program Files** の **Fonts** フォルダにフォントをコピーします。

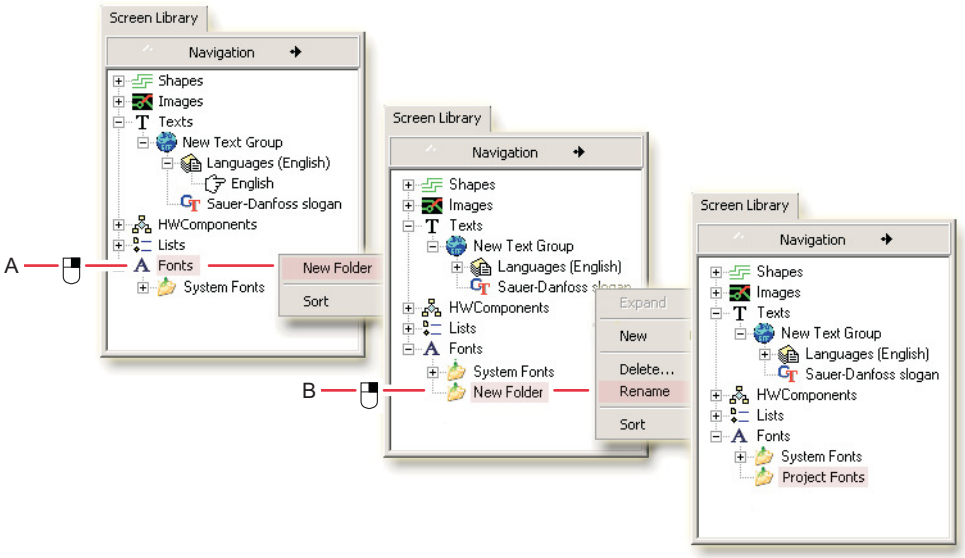
   A. **Setup** メニューで、**Install Font** をクリックして **Install Font** ウィンドウを開きます。

   B. **Install Font** ウィンドウで、**Select** をクリックして **Font** ウィンドウを開きます。

   C. **Font** ウィンドウで、任意の **Font**、 **Font style**、 **Size** を選択します。

      ローマン文字以外の文字を表示する場合、必要に応じて適切な Script タイプを選択します。

   D. **Font** ウィンドウと **Install Font** ウィンドウを閉じます。



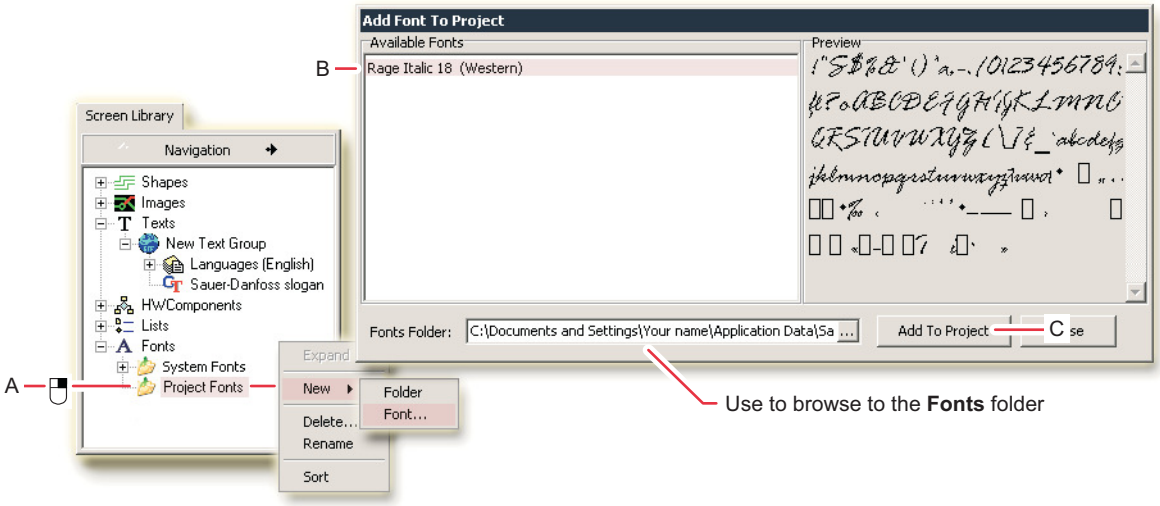
3. アプリケーションフォント用のフォルダを作成します。

   A. **Screen Library** タブで、**Fonts** カテゴリーを右クリックします。

   開いたポップアップウィンドウで、**New Folder** コマンドをクリックして新規フォルダを作成します。

   B. **New Folder** を右クリックします。

   開いたポップアップウィンドウで、**Rename** コマンドをクリックしてこのフォルダに **Project Fonts** など分かりやすい名前をつけます。



4. 追加したフォントをアプリケーションで使用できるようにします。

   このステップでは、**Fonts** フォルダにあるフォントを **Screen Library** タブの先ほど作成したフォルダにコピーします。

   A. **Project Fonts** フォルダを右クリックします。

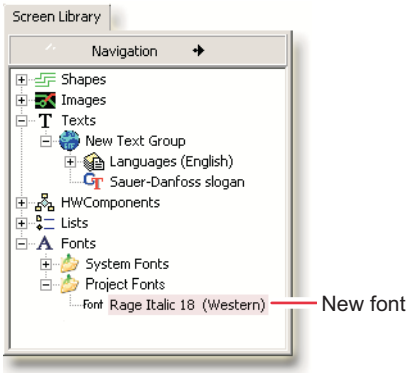
      開いたポップアップメニューで、**New > Font** コマンドをクリックして **Add Font to Project** ウィンドウを開きます。

   B. **Add Font to Project** ウィンドウで、追加したいフォントをクリックします。

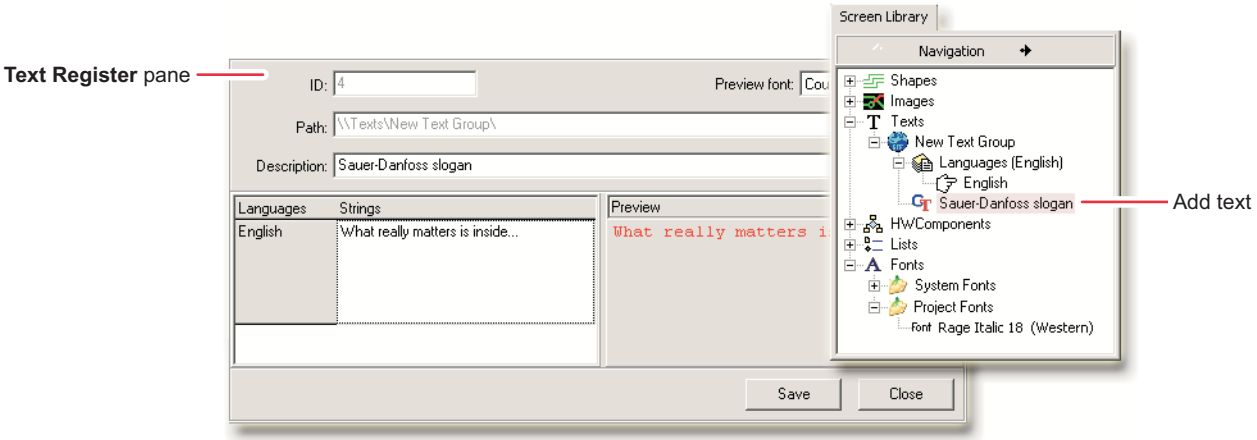
   C. **Add to Project** をクリックしてフォントを追加します。

   D. さらにフォントを追加するときは、この手順を繰返します。

   E. **Add Font to Project** ウィンドウを閉じます。

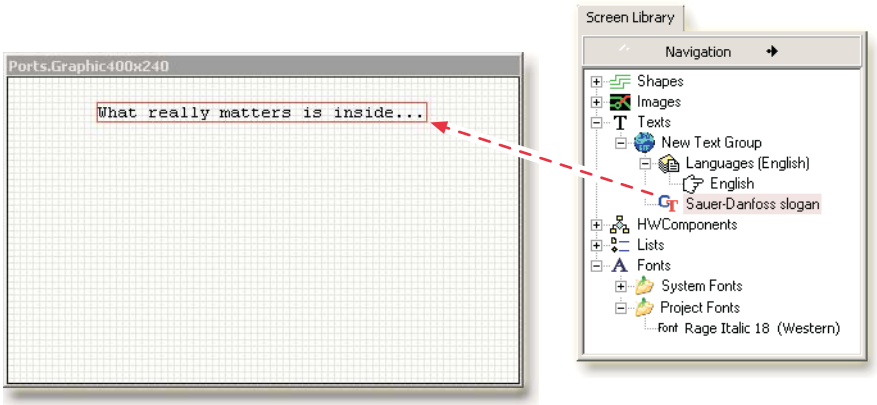


Screen Library タブに追加したフォントが表示されます。

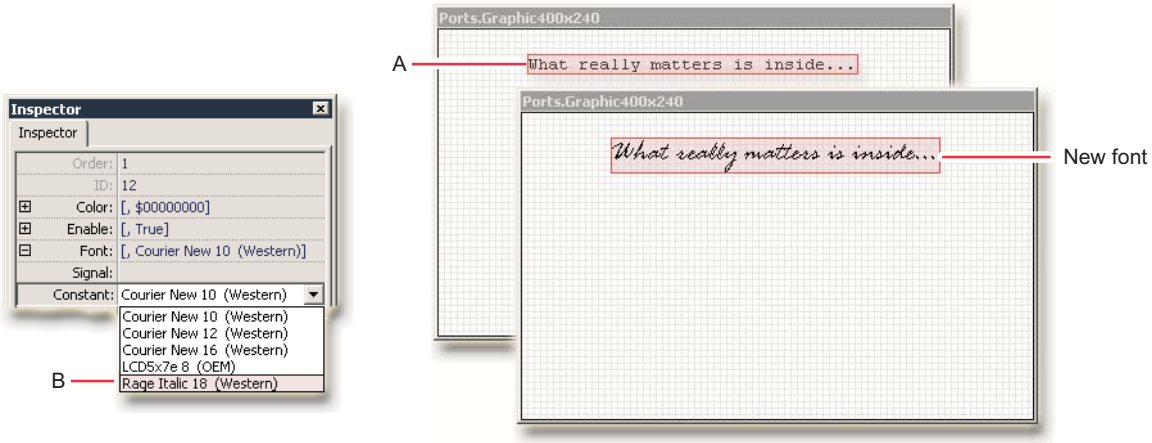


5. **Text Register** ペインを使用して、**Screen Library** タブの **Texts** ツリーにテキストを追加します。

   Text Register ペインの詳細については、*xxx*ページの「*Define Screen ページ/テキスト登録*」を参照してください。



6. テキストを **Screen Library** タブからレイアウトペインにドラッグします。



7. テキストの **Font** プロパティを変更します。

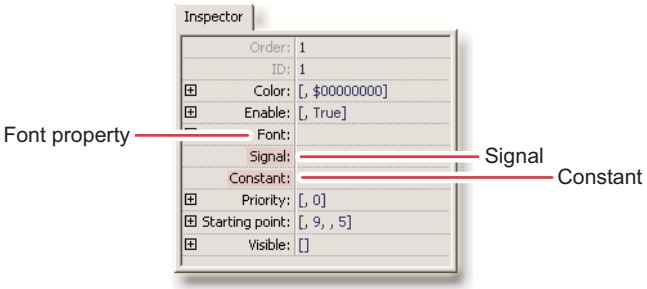
   A. Layout ペインでテキストをクリックします。

   B. **Inspector** タブで、テキストの **Font** プロパティを **Constant** システムフォントから新規アプリケーションフォントに変更します。

8. アプリケーションをコンパイルして、ディスプレイにダウンロードします。

Define Screen ページ / 文字列のフォントプロパティの切り替え

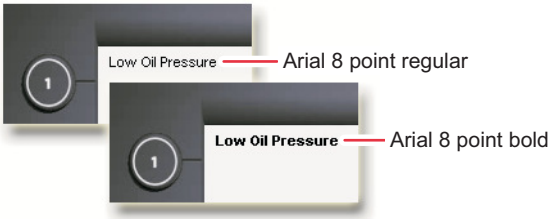
それぞれの文字列は **Font** プロパティを持ち、その書体（Arial など）、サイズ（8 ポイントなど）、スタイル（標準、斜体、太字など）を決めています。



Font プロパティは **Constant** または **Signal** 経由で設定することができます。

ここでは、文字列の **Font** プロパティを、フォントの書体、サイズ、スタイルを記述した **NameCodeGen** 値の入力を **Signal** を使って切り替える方法を説明します。

Signal を使って文字列の **Font** プロパティを切り替える一般的な用途としては、以下の図に示すように文字列を標準書体と太字で切り替えるなどがあります。



ここでは以下のことを前提とします。This how-to assumes that you have:

- **Define Areas** ブロックにハードウェアポートがアサインされていること。

  詳細については、*504*ページの*Define Areas Page/Assign a Hardware Port*を参照してください。

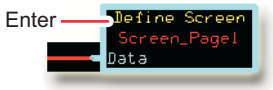
- **Define Screen** ブロックにスクリーンエリアがアサインされていること。

  詳細については、*511*ページの*Define Screen Page/Assign a Screen Area*を参照してください。

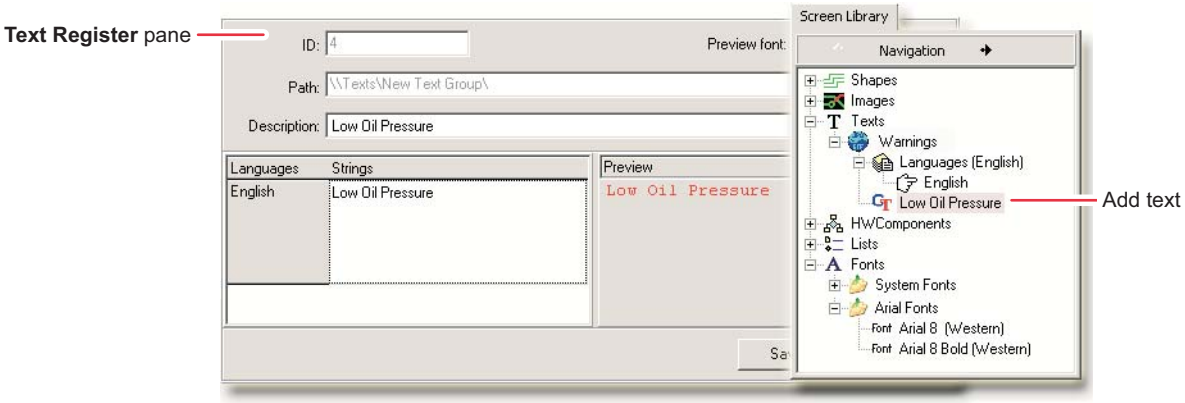
- 文字列に適用しようとするフォントがインストールされていること。

  （ここでは Arial regular 8 ポイントと Arial bold 8 ポイントを使用します。）

  詳細については、*537*ページの*Define Screen Page/Install Additional Fonts*を参照してください。.

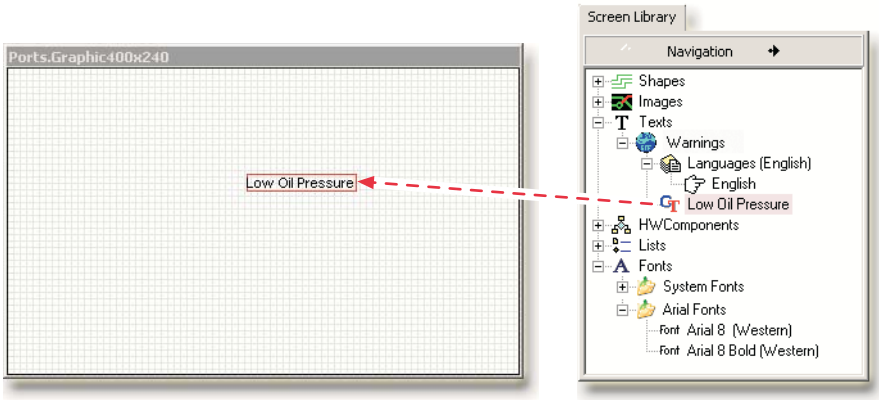


1. **Define Screen** ブロックに入ります。

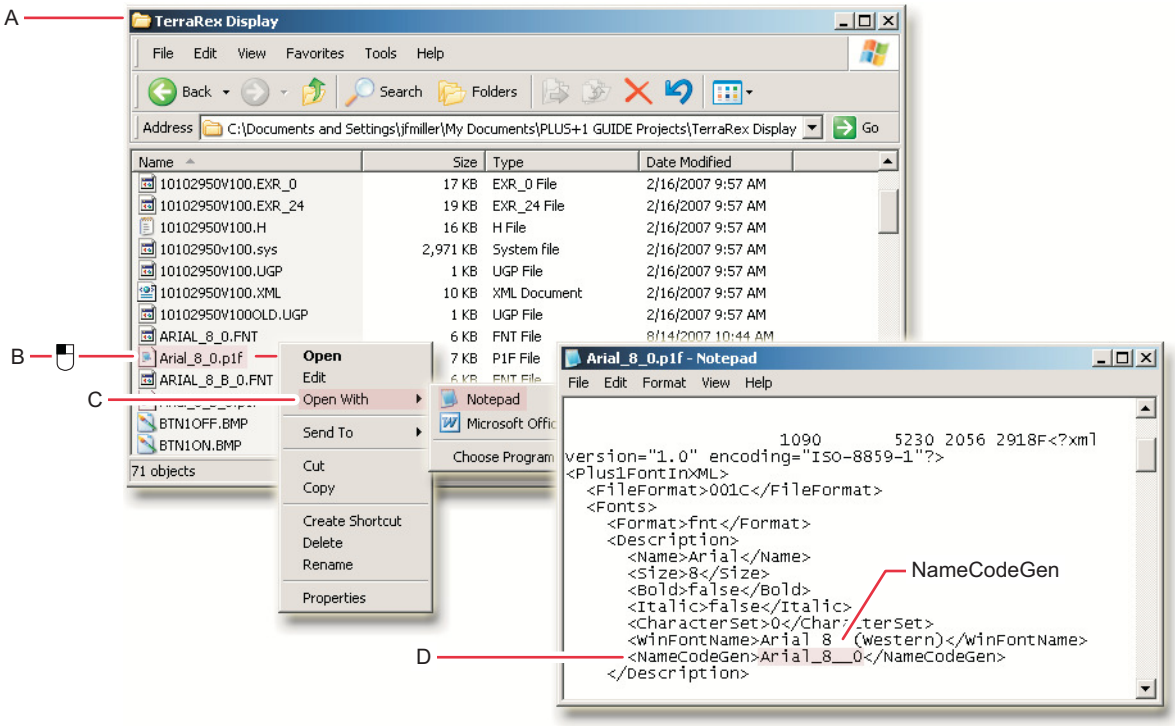


2. 文字列登録ペインを使って **Screen Library** タブの **Texts** ツリーに文字列を追加します。

   Text Register ペインの詳細については、*394* ページの「*Define Screen ページ/テキスト登録*」を参照してください。



3. **Screen Library** タブから文字列をレイアウトペインにドラッグします。



4. 文字列に適用しようとするフォントの **NameCodeGen** 値を見つけます。

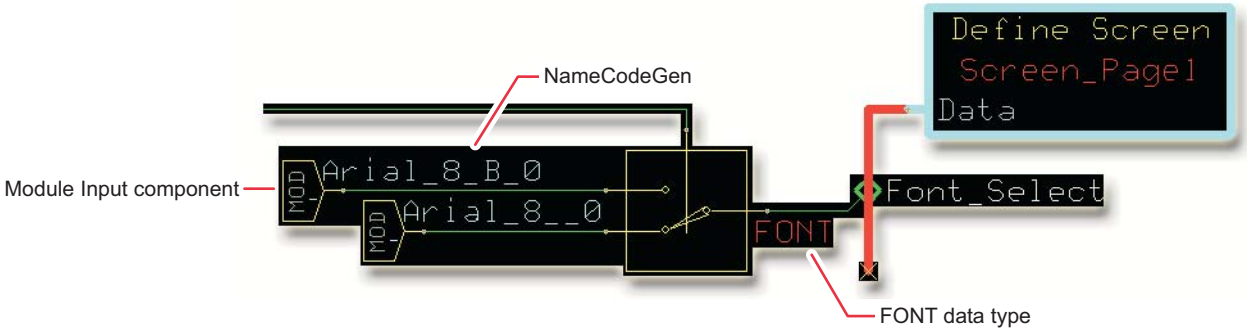
   A. プロジェクトフォルダを開いてブラウズします。

   B. PIF (PLUS+1 Font)ファイルを右クリックしてポップアップショートカットメニューを表示させます。

   C.**Open with** ショートカットメニューコマンドを使用して PIF ファイルをメモ帳で開きます。

   D. フォントの **NameCodeGen** 値を見つけます。

   E. 文字列に適用しようとしている他のフォントについても同様の手順で **NameCodeGen** 値を探します。



5. 文字列に適用する **NameCodeGen** 値を切り替えるロジックを作成します。

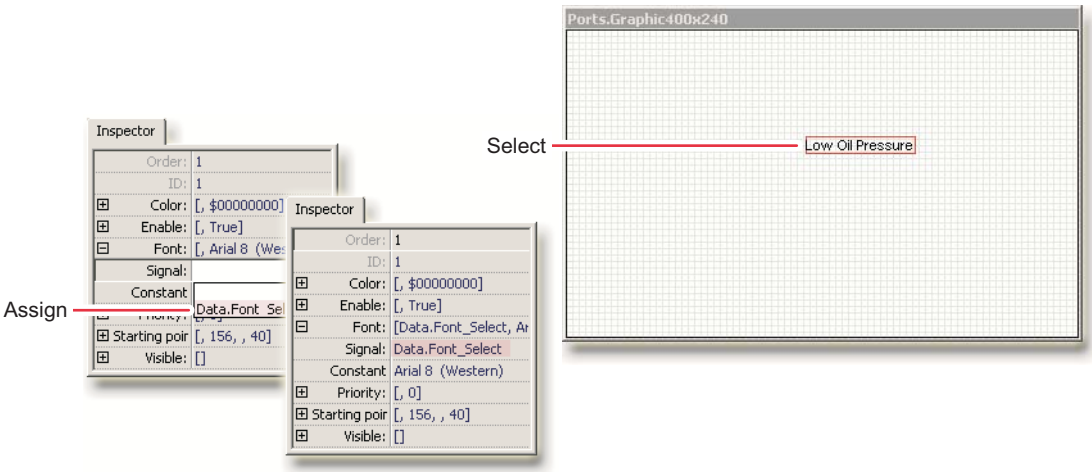
   When fully “wired” the logic of the signal flow in this figure is as follows:

   - 2 つの Single Wire **Module Input** コンポーネントが **NameCodeGen** 値を **Boolean Controlled Switch 2** コンポーネントに入力

     （**NameCodeGen** 値を入力するために Single-wire **Module Input** コンポーネントが必要です。他のロジックはアプリケーションによって異なります。）

   - **Boolean Controlled Switch 2** コンポーネントが以下の動作をします
     - **NameCodeGen** 値を **Arial_8_ _0** と **Arial_8_B_0** の間で切り替える。
     - **NameCodeGen** 値を持つ **Font_Select** 信号を **Define Screen** ページに出力する。**Font_Select** 信号のデータタイプは **Font** に設定する。
   - **Define Screen** ページ内で、**Font_Select** 信号が文字列の **Font** プロパティを設定します。

     Font_Select による **NameCodeGen** 値の設定値が、
     - **Arial_8_ _0** の場合、文字列は Arial 8 ポイント標準字体で表示される。
     - **Arial_8_B_0** の場合、文字列は Arial 8 ポイント太字体で表示される。



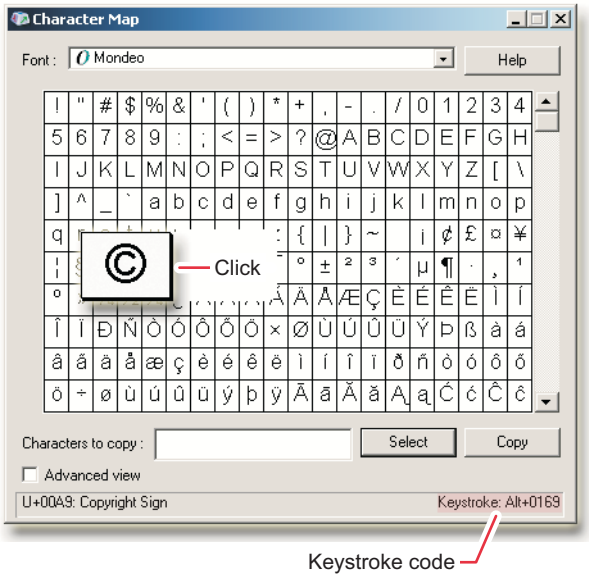
6. **Define Screen** ページ内で、選択した文字列の **Font** プロパティを制御するために **Signal** をアサインします。

Define Screen ページ / 特殊キャラクタの表示

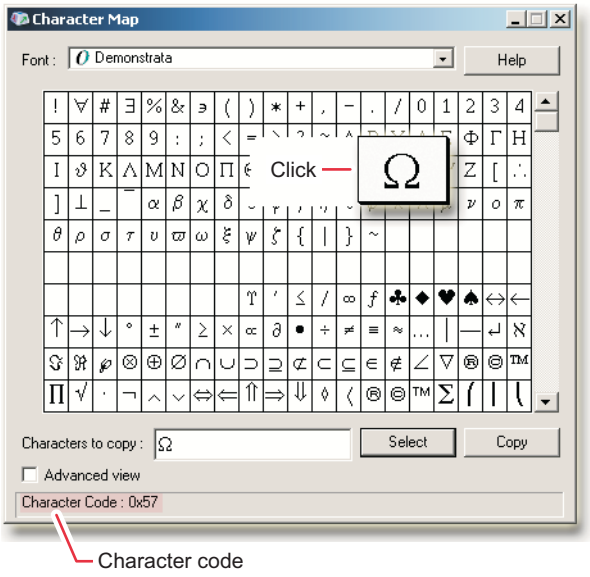
スクリーンエディタアプリケーションは±、©、Ωなどの特殊キャラクタを表示できます。これらのキャラクタは多くの Western TrueType や OpenType などのフォントセットに含まれていますが、キーボードから直接入力することができません。

特殊キャラクタを入力するには二つの方法があります。

- キーストロークコードによる方法：キーストロークコードが使える特殊キャラクタに使用
- キャラクタコードによる方法：キャラクタコードを持つがキーストロークコードを持たないキャラクタに使用



Use keystroke code method



Use character code method

以下の手順によってキーストローク法かキャラクタコード法のどちらを使うかを決定してください。

1. **Character Map** ウィンドウを表示します。

   Start > **Programs** > **Accessories** > **System Tools** > **Character Map**

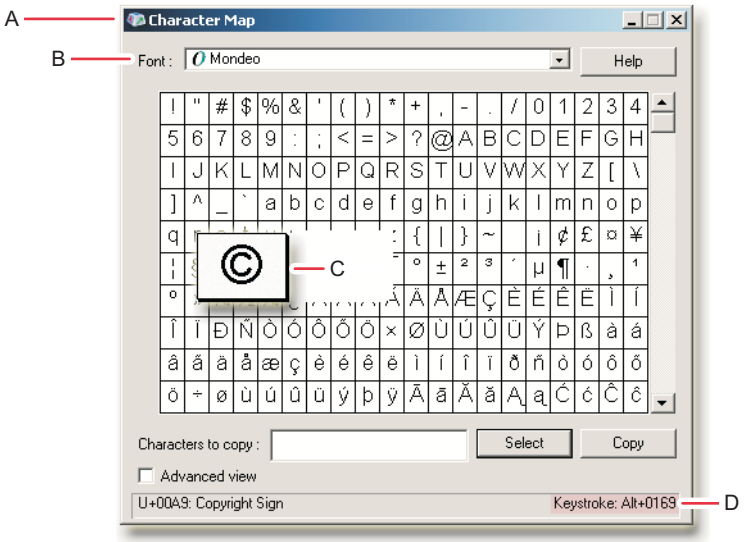
2. 特殊キャラクタの **Font** を選択します。
3. 特殊キャラクタをクリックします。
4. 入力方法を決定します。

   Character Map ウィンドウに、

   - キーストロークコードが表示された場合、***549***ページの***Keystroke Code Method***を使用してください。
   - キャラクタコードしか表示されない場合、***552***ページの***Character Code Method***を使用してください。

キーストロークコードによる方法

スクリーンエディタは 255 より大きなキーストロークコードを持つキャラクタを表示することはできません。

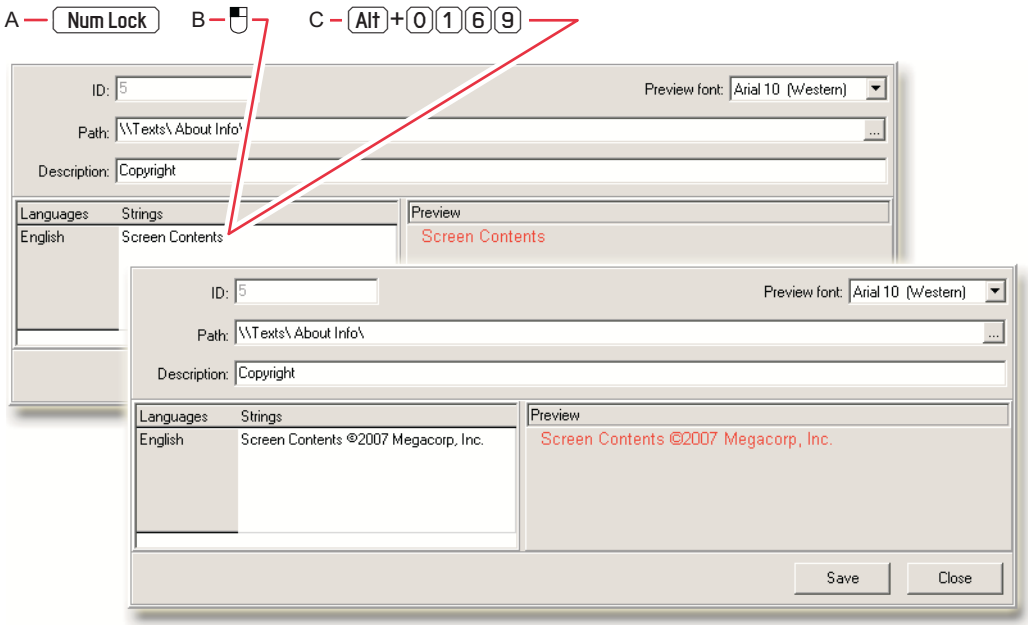


1. 特殊キャラクタのキーストロークコードを確定します。
   A. **Character Map**ウィンドウを表示させます。[1]

      Start > **Programs** > **Accessories** > **System Tools** > **Character Map**

   B. 特殊キャラクタの **Font** を選択します。
   C. 特殊キャラクタをクリックします。
   D. 特殊キャラクタの 4 桁のキーストロークコードをメモします。

[1] 日本語の Windows OS 環境ではこれ以外の方法で表示させることができる場合があります。



2. 文字列登録ペインで、特殊キャラクタを入力します。

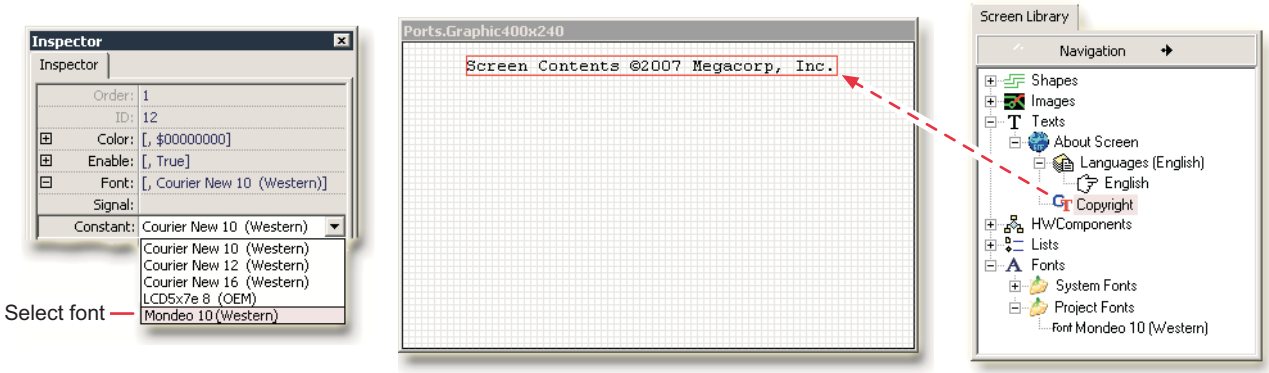
   A. キーボードの Num Lock がオンになっていることを確認します。

   B. 特殊キャラクタを表示させたいところをクリックします。

   C. Alt を押しながら 4 桁のキーストロークコードを入力します。

   キーストロークコードを入力するときはテンキーの数値キーボード(7/Home や 9/PgUp などのキー)を使用してください。

*551*ページの表 ***Keystroke codes for special characters***も参照してください。



3. 文字列を **Screen Library** タブからレイアウトペインにドラッグします。

   Inspector タブ内で文字列の **Font** プロパティを設定します。

この表の 4 桁キーストロークコードを用いて簡単に特殊キャラクタを入力することができます。

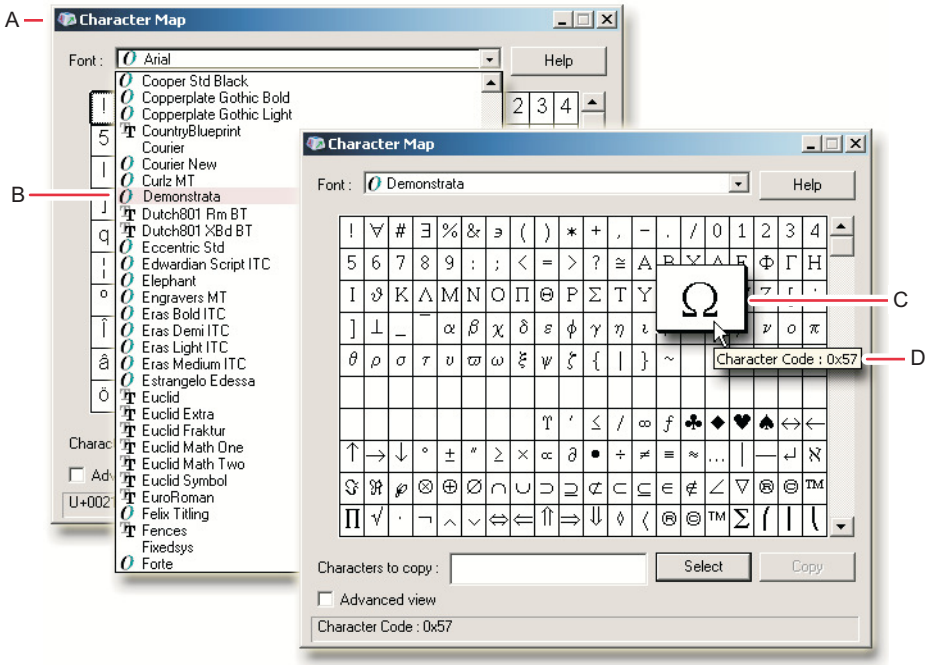
特殊キャラクタのキーストロークコード

キャラクタ	Num Lock+Alt+	キャラクタ	Num Lock+Alt+	キャラクタ	Num Lock+Alt+	キャラクタ	Num Lock+Alt+
*	0160	¾	0190	Ü	0220	ú	0250
¡	0161	¿	0191	Ý	0221	û	0251
¢	0162	À	0192		0222	ü	0252
£	0163	Á	0193	ß	0223	ý	0253
¤	0164	Â	0194	à	0224	þ	0254
¥	0165	Ã	0195	á	0225	ÿ	0255
¦	0166	Ä	0196	â	0226		
§	0167	Å	0197	ã	0227		
¨	0168	Æ	0198	ä	0228		
©	0169	Ç	0199	å	0229		
ª	0170	È	0200	æ	0230		
«	0171	É	0201	ç	0231		
¬	0172	Ê	0202	è	0232		
-	0173	Ë	0203	é	0233		
®	0174	Ì	0204	ê	0234		
¯	0175	Í	0205	ë	0235		
°	0176	Î	0206	ì	0236		
±	0177	Ï	0207	í	0237		
²	0178	Ð	0208	î	0238		
³	0179	Ñ	0209	ï	0239		
´	0180	Ò	0210	ð	0240		
µ	0181	Ó	0211	ñ	0241		
¶	0182	Ô	0212	ò	0242		
·	0183	Õ	0213	ó	0243		
¸	0184	Ö	0214	ô	0244		
¹	0185	×	0215	õ	0245		
º	0186	Ø	0216	ö	0246		
»	0187	Ù	0217	÷	0247		
¼	0188	Ú	0218	ø	0248		
½	0189	Û	0219	ù	0249		

*改行なしスペースの入力

キャラクタコードによる方法

スクリーンエディタは 16 進 FF（10 進 255）より大きなキャラクタコードを持つキャラクタを表示することはできません。



1. 特殊キャラクタの 16 進キャラクタコードを見つけます。

   A. **Character Map**ウィンドウを表示します。[2]

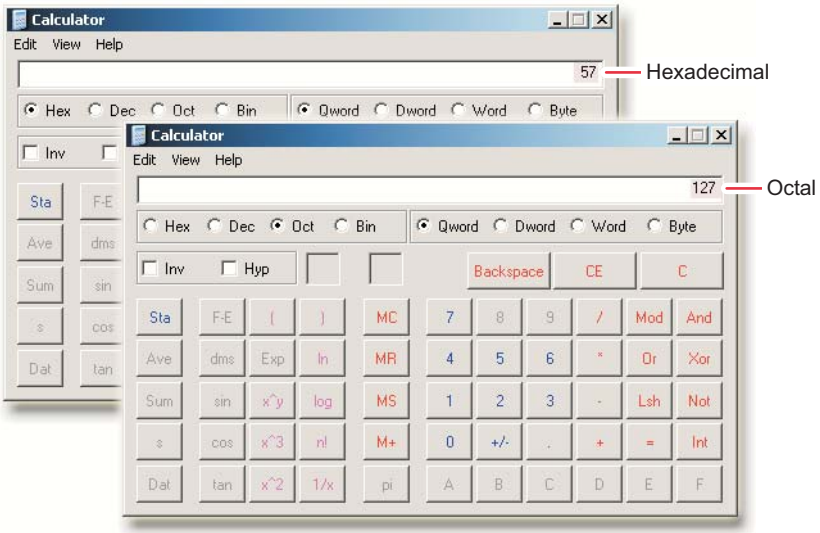
   Start > **Programs** > **Accessories** > **System Tools** > **Character Map**

   B. 特殊キャラクタの **Font** を選択します。

   C. 特殊キャラクタをクリックします。

   D. 特殊キャラクタの 16 進 **Character Code** をメモします。

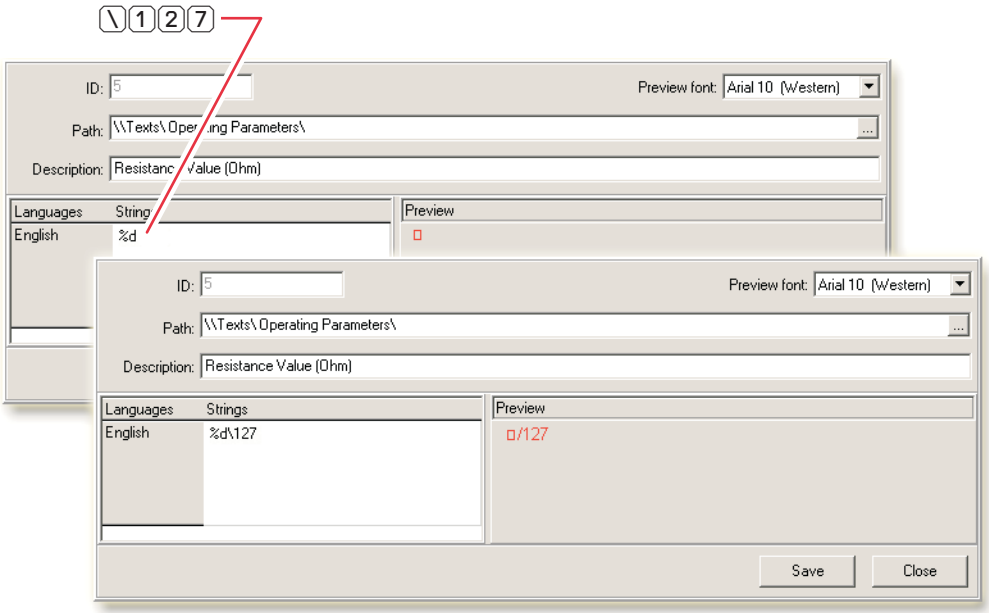
[2] 日本語の Windows OS 環境ではこれ以外の方法で表示させることができる場合があります。



2. 特殊キャラクタの **Character Code** を 16 進数から 8 進数に変換します。

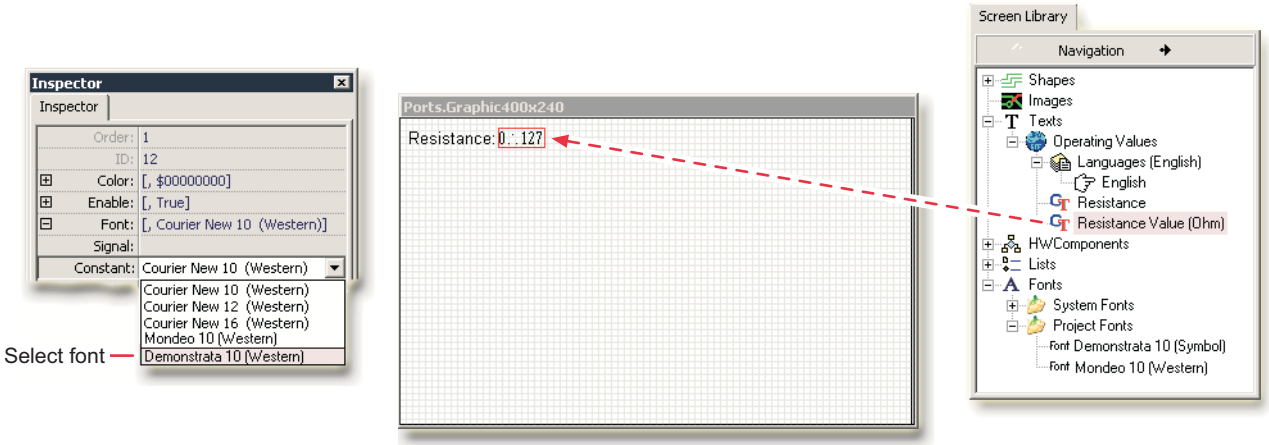
   Windows XP は 16 進キャラクタコードを使用しますが、スクリーンエディタは 8 進キャラクタコードを使用します。

   (XP 付属の **Calculator**(電卓)には 16 進数から 8 進数への変換機能が備わっています。)



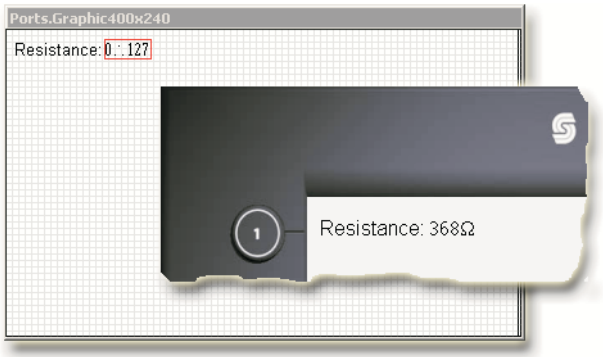
3. スクリーンエディタの文字列登録ペインで、バックスラッシュ(\)[3]に続けて特殊キャラクタの 8 進キャラクタコードを入力します。

[3] 日本語 Windows OS の場合はバックスラッシュ(\)の代わりに円記号(¥)を使う場合があります。



4. 文字列を **Screen Library** タブからレイアウトペインにドラッグします。

   Inspector タブで、文字列の **Font** プロパティを設定します。



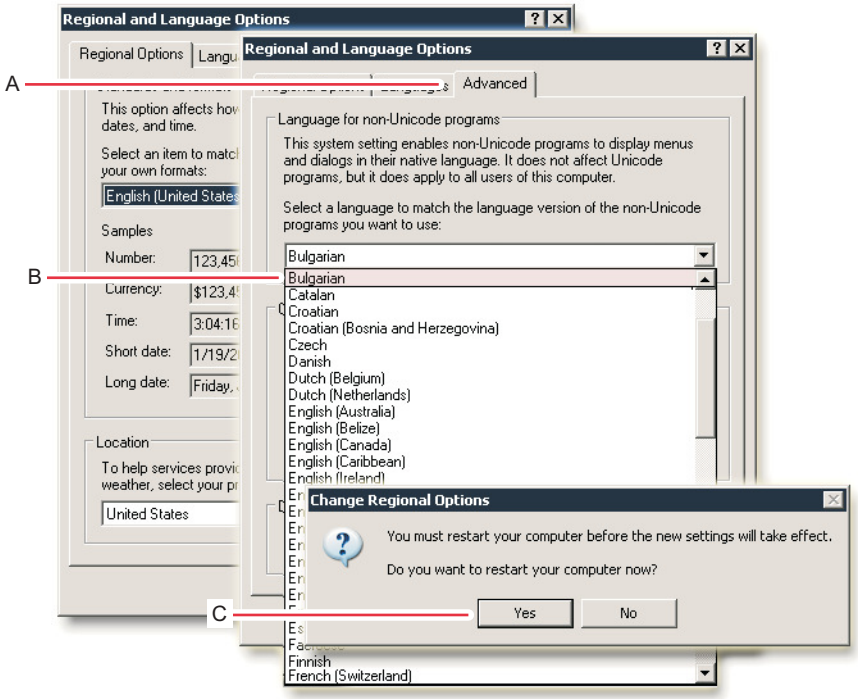
アプリケーションファイルをコンパイルし、グラフィックディスプレイにダウンロードすると特殊キャラクタが表示されます。

Define Screen ページ / 非ローマンアルファベットの表示

ザウアーダンフォス・ダイキンのグラフィックディスプレイは、キリル文字などの非ローマンアルファベットを表示させることができます。[4]

この手引きでは、以下の方法を示しています。

- 非ローマンアルファベットを Windows OS で入力できるように設定する方法
- 文字列登録ペインで非ローマンアルファベットを使用する言語で文字列を入力する方法



1. Windows OSで表示する言語を選択します。[5]

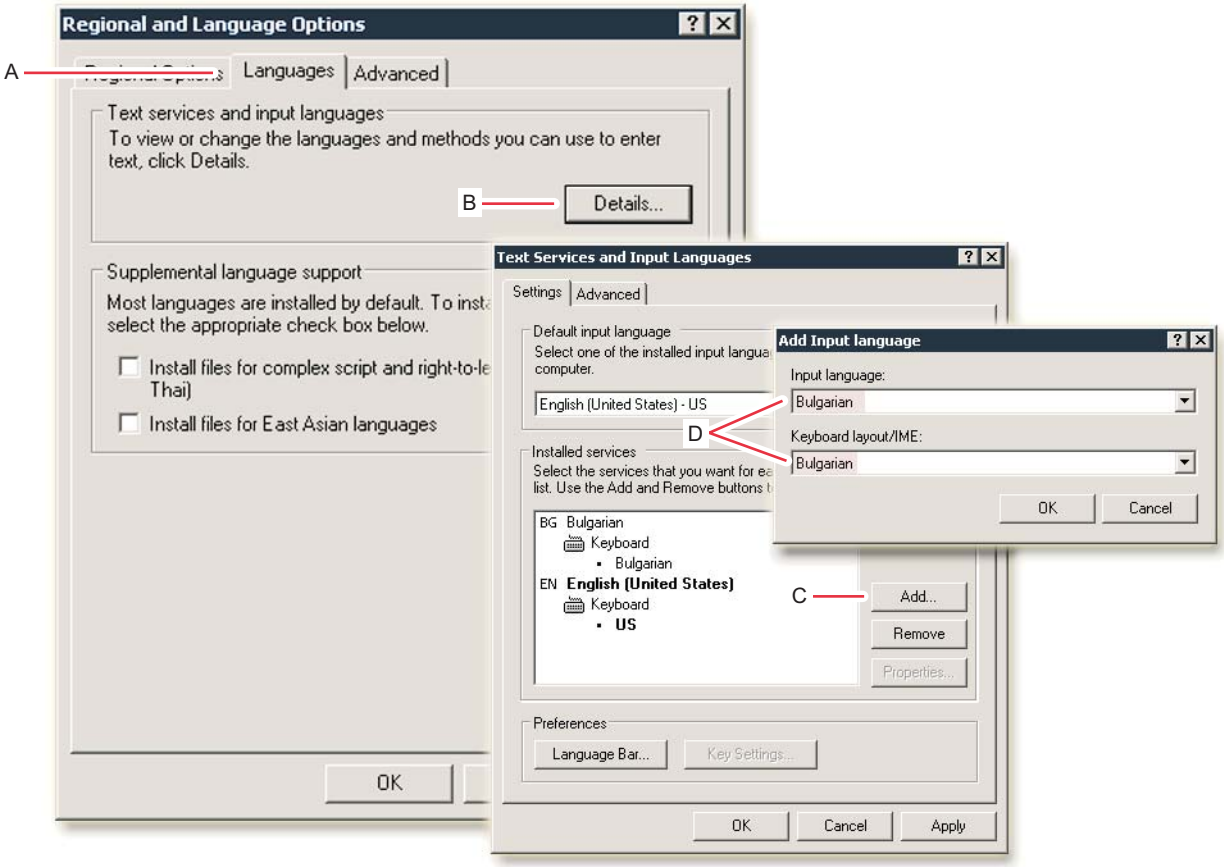
   A. **Regional and Language Options**（地域と言語オプション）ウィンドウの **Advanced** タブを表示します。

   パス: **Start** メニュー > **Settings** > **Control Panel** > **Regional and Language Options** ウィンドウ > **Advanced** タブ

   B. **Regional and Language Options** ウィンドウの **Advanced** タブで所望の言語を選択します。

   C. **Change Regional Options** ウィンドウが表示されるので、**Yes** をクリックしてコンピュータを再起動します。

[4] PLUS+1 GUIDE およびディスプレイ製品は日本語や中国語などのマルチバイト文字の表示には対応していません。
[5] ここでは英語版 Windows OS の場合の操作を示しています。



2. Windows OS で、ステップ 1 で選択した言語に適した **Input language** と **Keyboard layout/IME** を選択します。

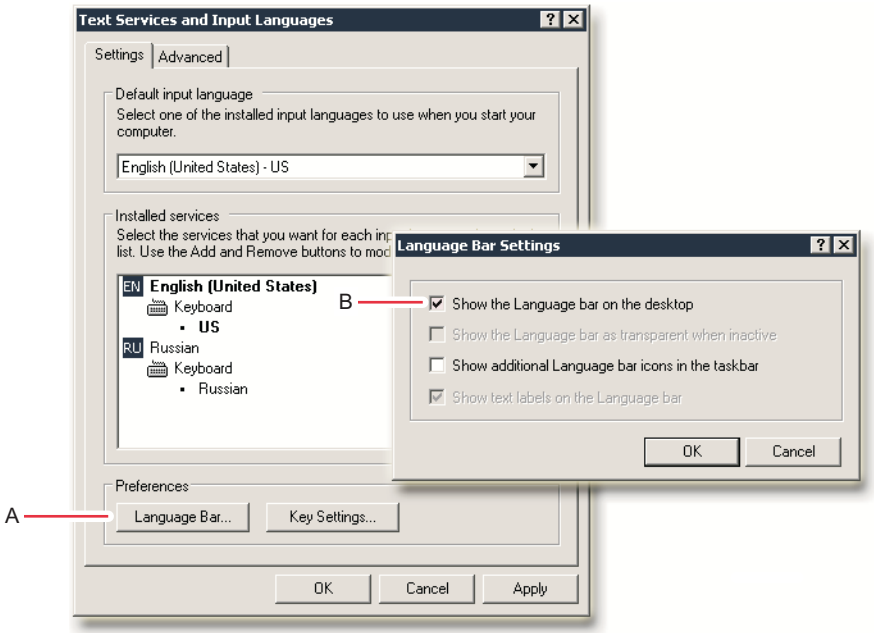
   A. **Regional and Language Options** ウィンドウの **Languages** タブを表示します。

      パス: **Start** メニュー > **Settings** > **Control Panel** > **Regional and Language Options** ウィンドウ > **Languages** タブ

   B. **Details** をクリックして **Text Services and Input Languages** ウィンドウの **Settings** タブを表示させます。

   C. **Text Services and Input Languages** ウィンドウで **Add** をクリックして **Add Input Language** ウィンドウを表示させます。

   D. **Add Input Language** ウィンドウで、適切な **Input language** と **Keyboard layout/IME** を選択します。



3. デスクトップに言語バーを表示させます。

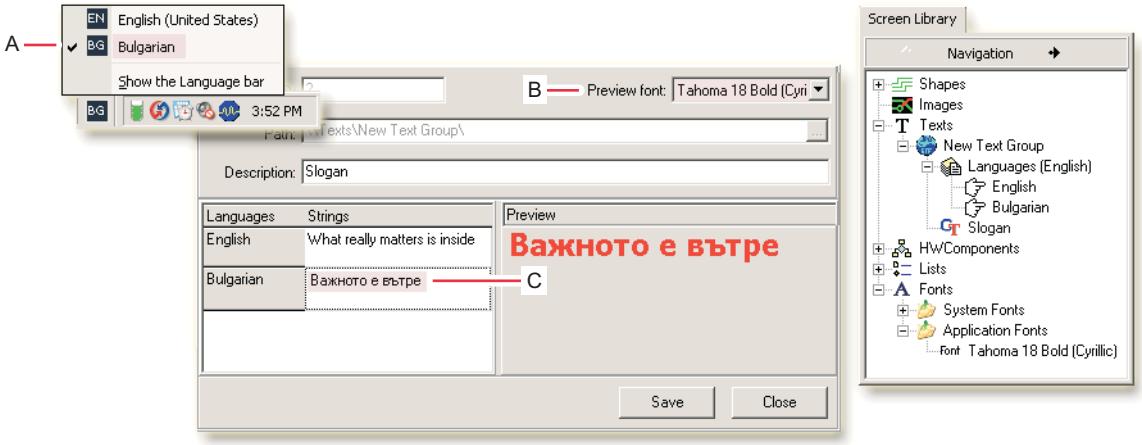
   A. **Text Services and Input Languages** ウィンドウで **Language Bar** をクリックして **Language Bar Settings** ウィンドウを表示させます。

   B. **Language Bar Settings** ウィンドウで、**Show Language bar on the desktop** をクリックします。

   C. **Language Bar Settings** ウィンドウと **Text Services and Input Languages** ウィンドウの両方を閉じます。

4. フォントをインストールするには、***537***ページの***Define Screen Page/Install Additional Fonts***を参照してください。

5. 必要なフォントをインストールしたら、次のステップに進みます。



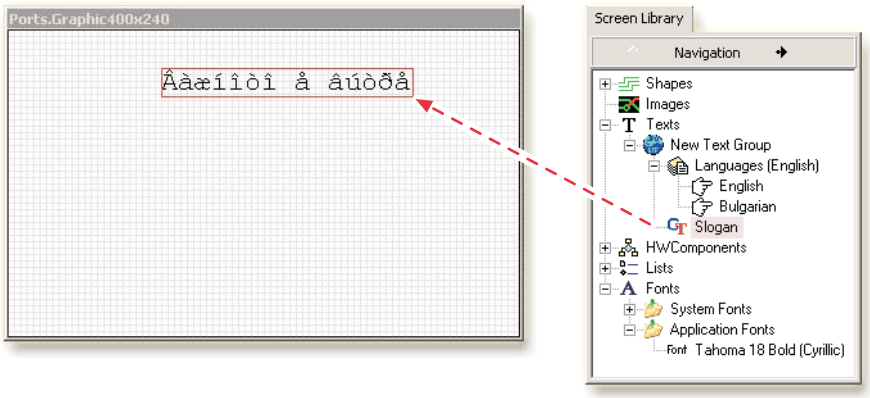
6. 文字列登録ペインで、文字列を **Screen Library** タブの **Texts** ツリーに追加します。

   A. 言語バーを使用して正しいキーボード入力を選択します。

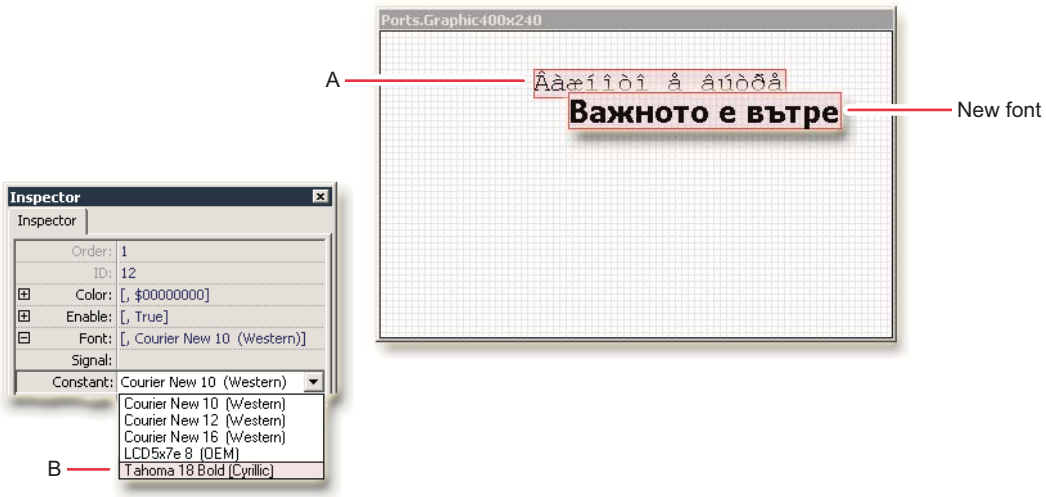
   B. 文字列登録ペインで、**Preview font** のプルダウンメニューを使って正しいプレビューフォントを選択します。

   C. 文字列を入力します。

   D. **Save** をクリックして変更を保存します。**Close** をクリックして文字列登録ペインを閉じます。



7. 文字列を **Screen Library** タブからレイアウトペインにドラッグします。



8. 文字列の **Font** プロパティを変更します。

   A. レイアウトペインで文字列をクリックします。

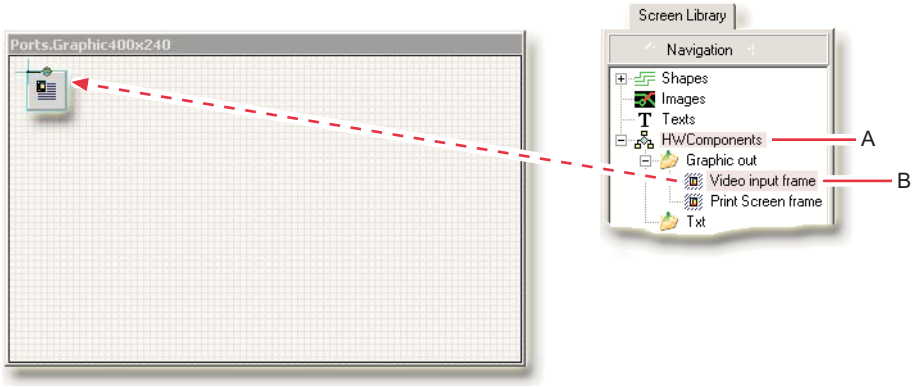
   B. **Inspector** タブで、文字列の **Font** プロパティの **Constant** をシステムプロパティから非ローマンアルファベット用のフォントに変更します。

9. プロジェクトをコンパイルしてディスプレイにダウンロードします。

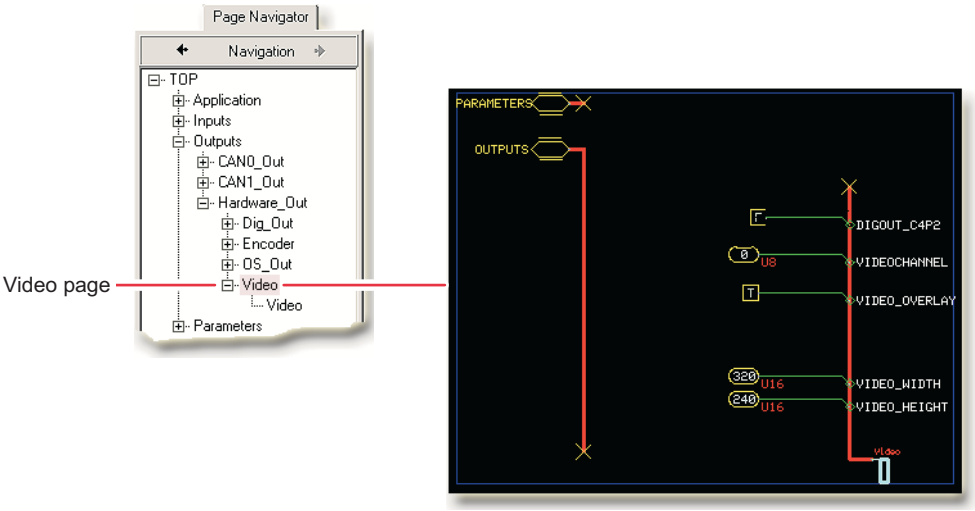
Define Screenページ/動画表示

必要なハードウェア： PAL カメラ、電源 12－24 VDC

1. **Define Screen** ページに入ります。
2. ポートとスクリーン･エリアをまだ割当てていなければ、割当を行います。

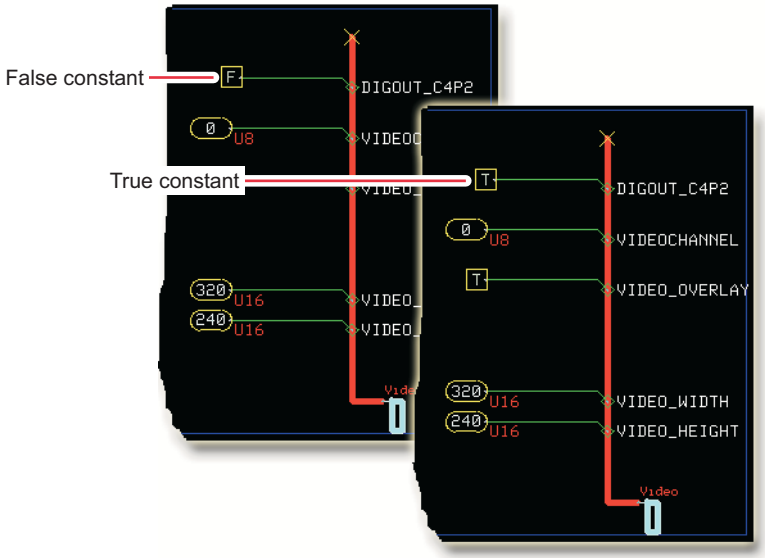


3. **Screen Library** タブの **Video input frame** をスクリーン･エリアにドラッグします。
   A. **Screen Libray** タブで、**HWComponents** ツリーをクリックして内容を表示します。
   B. **Video input frame** をスクリーン･エリアの左上隅にドラッグします。



4. **Screen Areas** ページを閉じて、**Video** ページを開きます。

   （**Video** ページは **Outputs** ページの中にあります。）



5. **Video** ページで、**DIGOUT** の **F**（False）定数を **T**（True）定数に置き換えてビデオ出力を有効にします。
6. アプリケーションをコンパイルして、ディスプレイにダウンロードします。
7. PAL カメラ出力をディスプレイに接続します。

Define Screenページ/画面印刷

画面印刷で次のことが可能になります。

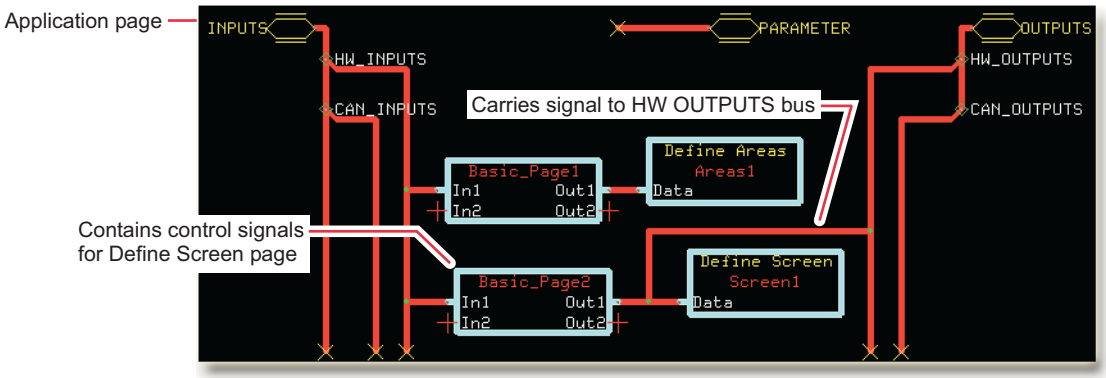
- ディスプレイ画面をキャプチャーして、その画面をディスプレイの RAM に保存します。
- キャプチャー画面を表示します。

  画面はディスプレイの電源を切るか、他の画面をキャプチャーするまで、ディスプレイのメモリに残ります。

画面印刷を実行するには 2 つの信号が必要です。

- 画面イメージをディスプレイの RAM に保存させるブール信号。以下の手順での **Prnt Scrn**（Print Screen）信号がこの機能を果たします。
- 保存された画面イメージをディスプレイに表示させるブール信号。以下の手順での **Dsp Prnt Scrn**（Display Print Screen）信号がこの機能を果たします。

  以下の手順では、ディスプレイのボタン **1**、**2** の入力がこれらの 2 つの信号を True にします。（アプリケーションによっては、別の入力を用いてこれらの信号を True にすることもあります。）

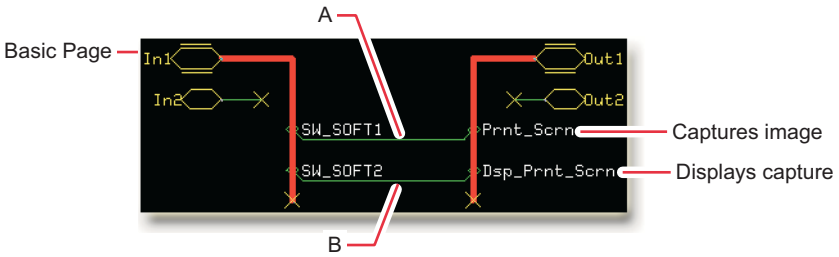


1. スクリーンエディタを使用して、ディスプレイのアプリケーションを作成します。

   このアプリケーションには次のものが必要です。

   - **Define Screen** ページの制御信号を含む **Basic Page**。
   - **Basic Page** から **HW Outputs** バスに信号を搬送するバス。

2. **Define Screen** ページの制御信号を含む **Basic Page** に入ります。



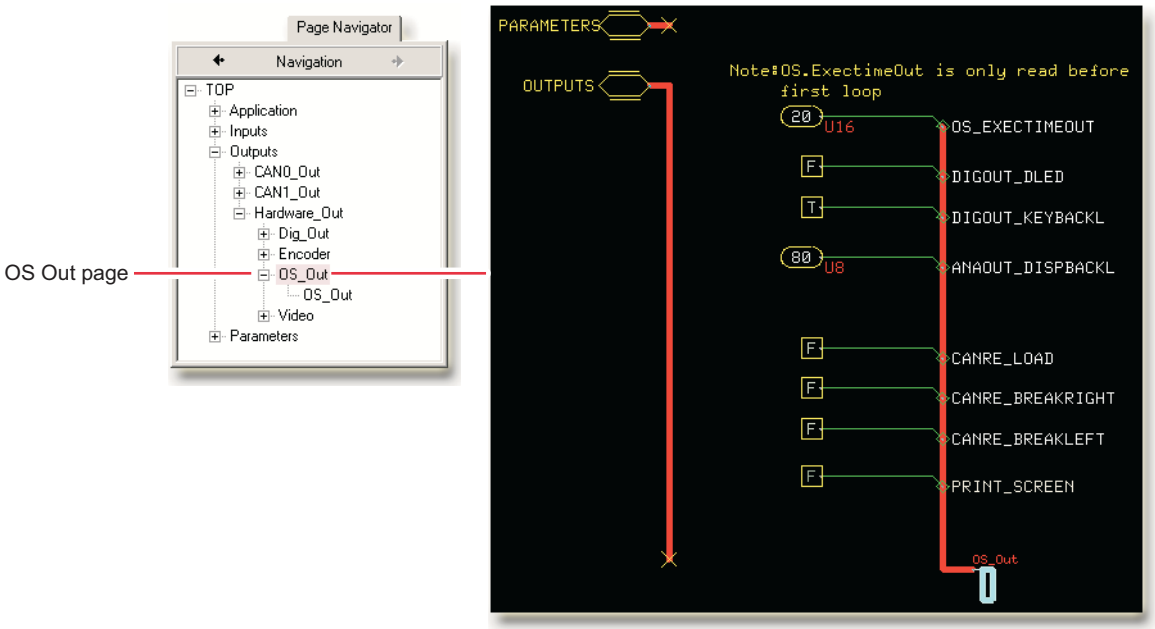
3. **Define Screen** ページの **Basic Page** で、画面イメージをキャプチャーする信号とキャプチャーした画面イメージを表示する信号を結線します。

A. **Prnt Scrn** 信号を結線して名前をつけます。

この信号が True のとき、ディスプレイに画面イメージをキャプチャーし、そのイメージを RAM に保存させます。

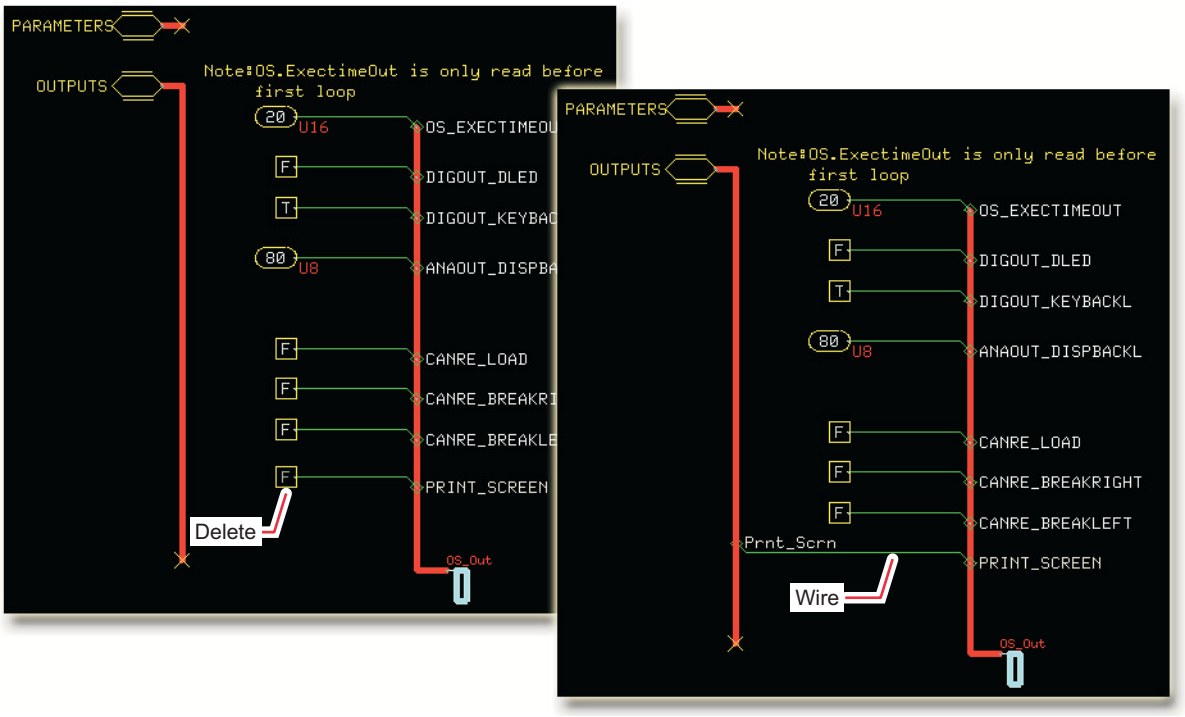
B. **Dsp Prnt Scrn** 信号を結線して名前をつけます。

この信号が True のとき、キャプチャーした画面イメージをディスプレイに表示させます。



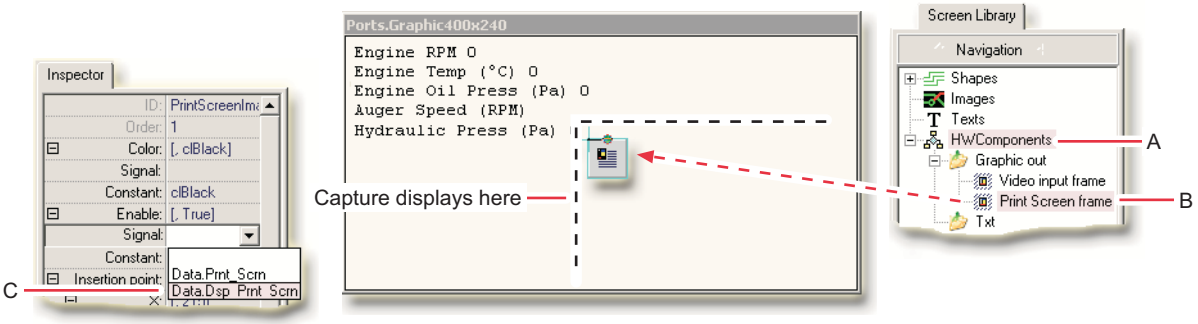
4. **Basic Page** を閉じて、**OS Out** ページを開きます。

（**OS Out** ページは **Outputs** ページの中にあります。）



5. **OS Out** ページで、**Prnt Screen** 信号を **PRINT SCREEN** 出力に結線します。

6. **Define Screen** ページに入ります。



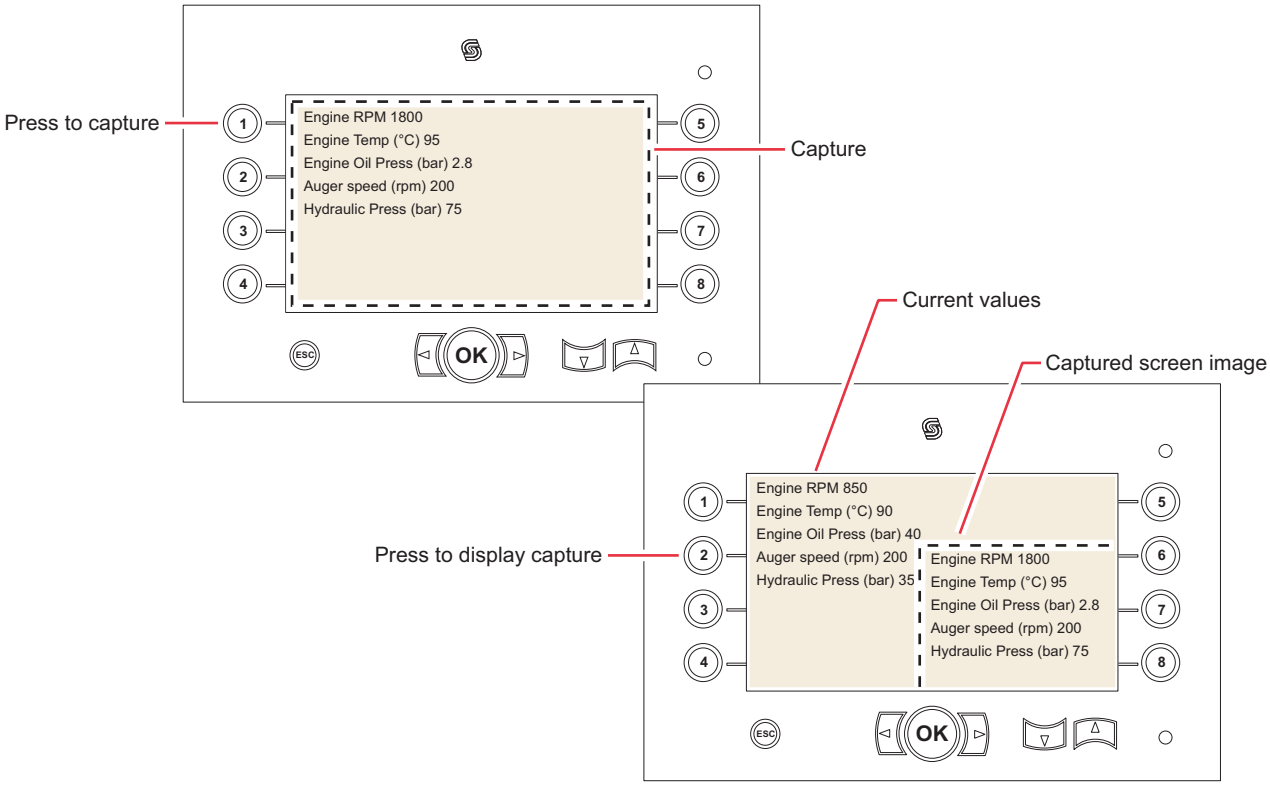
7. **Screen Library** タブの **Print Screen frame** をレイアウトペインにドラッグして、フレームを有効にする信号を選択します。

   A. **Screen Libray** タブで、**HWComponents** ツリーをクリックして内容を表示します。

   B. **Print Screen frame** をレイアウトペインにドラッグします。**Print Screen frame** を配置する場所により、キャプチャーした画面の左上隅位置が設定されます。

   C. **Inspector** タブで **Dsp Prnt Scrn** をクリックして、**Dsp Prnt Scrn** 信号によるキャプチャー画面の表示を **Enable** します。

8. アプリケーションをコンパイルして、ディスプレイにダウンロードします。



9. 画面イメージをキャプチャーして、キャプチャーした画面イメージを表示します。

   A. ディスプレイのボタン **1** を押して、**Prnt Scrn** 信号を True にします。

   ディスプレイが画面イメージをキャプチャーして、RAM に保存します。

   B. ディスプレイのボタン **2** を押して、**Dsp Prnt Scrn** 信号を True にします。

   キャプチャーされた画面イメージがディスプレイに表示されます。キャプチャー画面の表示位置は、**Prnt Screen frame** を配置した場所により決まります。

キーボードショートカット

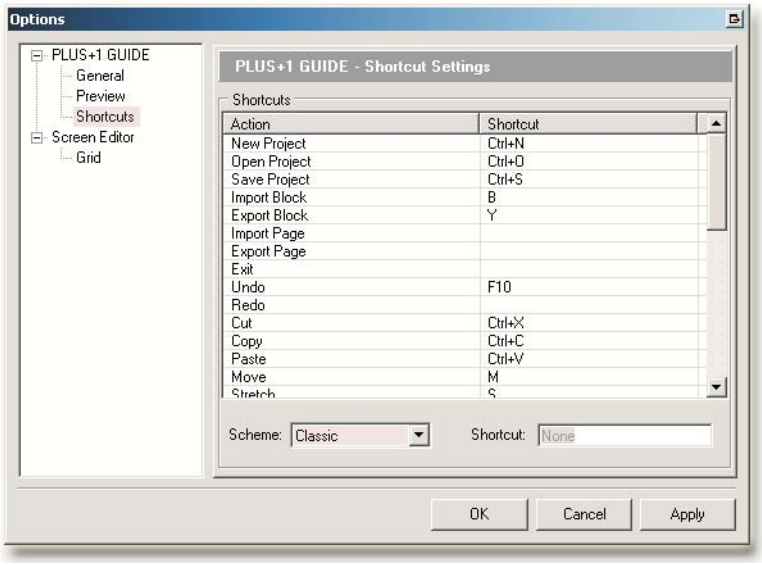
Classic キーボードショートカット体系がデフォルトのショートカット体系です。この章では **Classic** キーボードショートカットについて記述します。

この章の内容

キーボードショートカット 566
Classic キーボードショートカット体系について 567
基本キーボードショートカット 568
応用キーボードショートカット 569

Classicキーボードショートカット体系について

Classic キーボードショートカット体系はデフォルトのショートカット体系です。この章では **Classic** キーボードショートカットについて記述します。



Options ウィンドウを使ってキーボードショートカット体系を選択します。

PLUS+1 GUIDE ウィンドウ > **Setup** メニュー > **Options** ウィンドウ > **Shortcuts**

基本キーボードショートカット

基本キーボードショートカット

メニュー	コマンド	キー
File	New Project（新規プロジェクト）	c+n
	Open Project（プロジェクトを開く）	c+o
	Save Project（プロジェクトの保存）	c+s
	Import Block（ブロックのインポート）	b
	Export Block（ブロックのエクスポート）	y
Edit	Undo（操作の取り消し）	0
	Cut to Clipboard（カット）	c+x
	Copy to Clipboard（コピー）	c+c
	Paste from Clipboard（ペースト）	c+v
	Move（移動）	m
	Stretch（ストレッチ）	S
	Delete（削除）	X
	Copy/Repeat（コピー／繰り返し）	c
	Query/Change（クエリー／変更）	q
View	Enter Page（ページに入る）	e
	Leave Page（ページから出る）	l
	Refresh（再描画）	4
	Set Center（中央の設定）	i
	Recenter View（ビューの中央の再設定）	c+3
	Zoom In（ズームイン）	M
	Zoom Out（ズームアウト）	W
	Area（範囲）	3
	Fit Page（ページに合わせる）	h
Setup	Toggle Display Grid（グリッド切り替え）	G
Add	Route Wire/Bus（ワイヤ／バスの配線）	R
Tools	View Logical Net（論理ネット表示）	v
Help	PLUS+1 GUIDE Help（ヘルプ）	1

応用キーボードショートカット

応用キーボードショートカット

メニュー	コマンド	キー	機能
Edit	**Move（移動）**	6	選択したアイテムを反時計方向（CCW）に 1°単位で回転します
		7	選択したアイテムを反時計方向（CCW）に 45°単位で回転します
		8	選択したアイテムを 180°反転します
		9	選択したアイテムを反時計方向に 90°単位で回転します
		+	ページ上で接触していない別個の項目を移動します。 これらのキーを押しながら、移動したい各項目を選択します。キーを放します。 移動するときは選択項目のひとつをクリックします。
		-	+で選択した項目の選択を解除します。 このキーを押しながら、マウスをドラッグして各項目の選択を解除します。
		.	一群の選択項目のなかで、最後に加えた項目を移動します。 このキーを押しながらドラッグして、最後に加えた項目を移動します。
	Stretch（ストレッチ）	+	ページ上で接触していない別個の項目をストレッチします。 これらのキーを押しながら、ストレッチしたい各項目を選択します。キーを放します。 ストレッチするときは選択項目のひとつをクリック＆ドラッグします。
		-	+で選択した項目の選択を解除します。 このキーを押しながら、マウスをドラッグして項目の選択を解除します。
Edit	**Delete（削除）**	+	ページ上で接触していない別個の項目を削除します。 これらのキーを押しながら、削除したい各項目を選択します。キーを放します。 削除するときは選択項目のひとつをクリックします。表示される **Define** ウィンドウか **Select Item Class** ウィンドウで、**OK** をクリックします。
		-	+で選択した項目の選択を解除します。 このキーを押しながら、マウスをドラッグして各項目の選択を解除します。
		.	一群の選択項目のなかで、最後に加えた項目を削除します。 このキーを押しながらドラッグして、最後に加えた項目を反転表示します。 表示される **Attributes** ウィンドウで、**OK** をクリックして項目を削除します。
		x	ルート上で選択したセグメントを削除するには、セグメント選択中にこのキーを押します。
	Copy/Repeat（コピー／繰り返し）	+	ページ上で接触していない別個の項目をコピーします。 これらのキーを押しながら、コピーしたい各項目を選択します。キーを放します。 コピーするときは選択項目のひとつをクリックします。
		-	+で選択した項目の選択を解除します。 このキーを押しながら、マウスをドラッグして各項目の選択を解除します。
		.	一群の選択項目のなかで、最後に加えた項目をコピーします。 このキーを押しながらドラッグして、最後に加えた項目をコピーします。

応用キーボードショートカット

メニュー	コマンド	キー	機能
	Route	9	押すと未接続ルートへのセグメントの追加を停止します。
		s	押すと、90° および 180° 方向のみに配線を制限します。もう一度押すと、制限が解除されます。
		h	ルートに接続されているアイテムをハイライト表示します。 ルートに接続されているアイテムをハイライトするには、ルートを選択中にこのキーを押します。 **Edit** メニュー > **Remove Highlighting** コマンドを使用してハイライト表示を終了します。
		j	ルーティング中に押すと、ルート幅を変更できるウィンドウが表示されます。
		K	ルーティング中に押すと、バス配線とワイヤ配線をトグル切換します。
		n	ルート選択中に押すと、ルート情報を提供する **Nodes** ウィンドウが表示されます。
		n	押すと、選択したルートへ頂点の追加を開始します。 頂点の追加を止めるときは、9 を押します。
		n then x	ルート選択中に n を押します。その後、x を押してルートから頂点を削除します。 頂点の削除を止めるときは、9 を押します。

ページレイアウトガイドライン

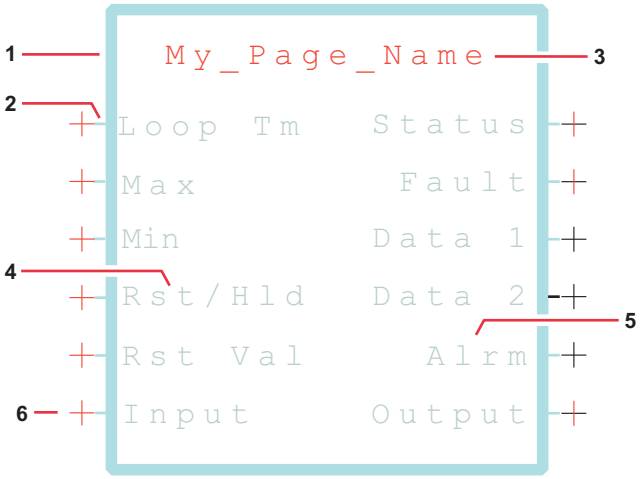
本章はページのレイアウトとポートのラベリングを行うための手引きです。

この章の内容

ページレイアウトガイドライン 571
ページガイドライン.....572
ポートラベルの略語 574
ポートラベル単位の略語 575

ページガイドライン

以下の図表にレイアウトのガイドラインを示します。ページインタフェースエディタでページビューを作成するときに利用してください。



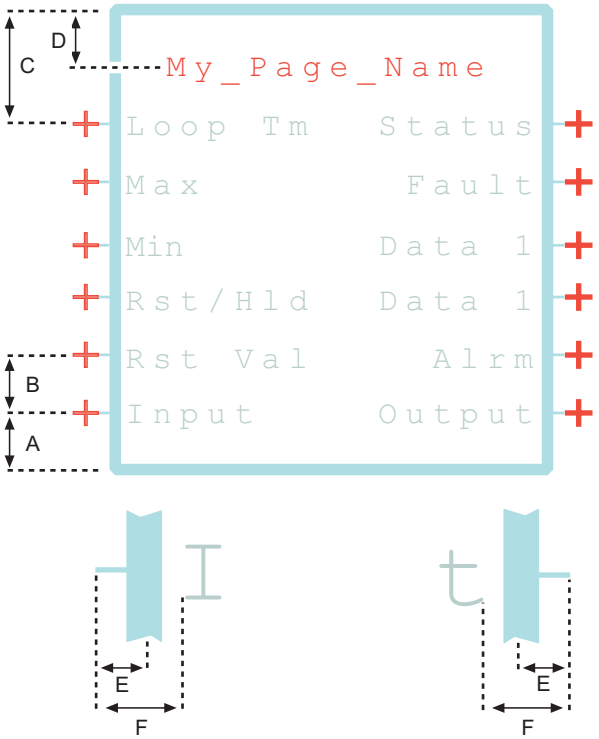
ページビューのレイアウトガイドライン

項目	Element（エレメント）	Layer/Color（階層/色）	Alignment（配置）	Other（その他）
1	境界線	SymbolBodyThin	———	Width: 1.0 mm
2	ピン	SymbolBodyThin	———	Style: Solid（実線）
3	ページ名	PageName	Center on block	Title Caps（各単語の先頭の文字だけ大文字）。語間スペースにアンダーライン（_）を引く。ファンクションの境界線を越えないこと。
4	入力ラベル	CoverSheet3	X left, Y center	Title Caps。語間スペースにアンダーライン（_）を引かない。
5	出力ラベル	CoverSheet3	X right, Y center	Title Caps。語間スペースにアンダーライン（_）を引かない。
6	挿入点	———	———	常に左下

Leave the **Layer** value of the **@PAGENAME** placeholder set to the **Layer** value of **PageName**. Changing this value may cause the page name to disappear. It can also cause compile problems and problems with the PLUS+1 program's advanced features.

文字サイズガイドライン

項目	Font（フォント）	Cell Height（セル高さ）	Cell Width（セル幅）	Stroke Width（ストローク幅）	Relative Size（相対サイズ）
ページ名	Standard System	2.5000	1.6664	0.3750	100 %
入力ラベル	Standard System	2.5000	1.6664	0.3750	100 %
出力ラベル	Standard System	2.5000	1.6664	0.3750	100 %



Detail: Input label and connection

Detail: Output label and connection

トップビュー寸法

項目	寸法	項目	寸法	項目	寸法
A	5.0 mm	C	10.0 mm	E	2.5 mm
B	5.0 mm	D	3.75 mm	F	3.75 mm

ポートラベルの略語

ポートラベルの略語（2 wheel～Right）

ラベル	略語	ラベル	略語	ラベル	略語
2 wheel	2Whl	Equal	Eq	Maximum	Max
4 wheel	4Whl	Error	Err	Middle, Mid	Mid
Accelerate	Acel	Fail	Fail	Minimum	Min
Alarm	Alrm	Fault	Flt	Negative	Neg
All	All	Feedback	Fdbk	Neutral	Neut
Angle	Ang	Forward	Fwd	Not	Not
Average	Avg	Found	Fnd	Number	Nmbr
Buzzer	Buzz	Four wheel	4Whl	Operator	Op
Bypass	Bpas	Frequency	Freq	Output	Out
Calibrate	Cal	Ground	Gnd	Pack	Pck
Clockwise	CW	Guard	Grd	Parameters	Para
Command	Cmd	Handle	Hndl	Pass	Pass
Control, Controller	Ctrl	High	Hi	Passive	Pasv
Coordinate	Cord	Hold	Hld	Point	Pt
Counterclockwise	CCW	Horn	Hrn	Positive	Pos
Deadband	Dbnd	Hysteresis	Hyst	Potentiometer	Pot
Decelerate	Dcel	Increase, Increment	Inc	Presence	Prs
Decrease, Decrement	Dec	Initialize	Init	Propel	Prpl
Default	Def	Input*	In	Pulse	Puls
Diameter	Dia	Left	L	Pulse pickup unit	PPU
Digital	Dig	Left front	LF	Pulse width modulation	PWM
Direction	Dir	Left rear	LR	Range	Rnge
Displacement	Disp	Length	Lgth	Read	Rd
Done	Done	Light Emitting Diode	LED	Reset	Rst
Down	Dn	Loop	Lp	Reverse	Rvs
Drive	Drv	Loop Time	LpTm	Revolution	Rev
Enable	Enbl	Low	Lo	Right	R

ポートラベルの略語（Right front～Write）

ラベル	略語	ラベル	略語	ラベル	略語
Right front	Rf	Solenoid	Sol	Time	Tm

Right rear	Rr	Speed	Spd	Value	Val
Sample	Smpl	Start	Strt	Wheel	Whl
Select	Slct	Status	Stat	Width	Wdth
Sensor	Snsr	Steer, Steering	Str	Work	Wrk
Set	Set	Stop	Stop	Write	Wr
Setpoint	Stpt	Sweep	Swp		
Soft	Sft	Switch	Sw		

ポートラベル単位の略語

単位の略語

項目	略語	項目	略語
ampere （アンペア）	**A**	**miles per hour （マイル毎時）**	**mph**
centimeter （センチメータ）	**cm**	**millimeter （ミリメートル）**	**mm**
connector （コネクタ）	**c**	**minute (time) （分）**	**min**
degree （度）	**Deg**	**ohm （オーム）**	**ohm**
degree Celsius （セ氏）	**C**	**pascal （パスカル）**	**Pa**
degree Fahrenheit （華氏）	**F**	**pin （ピン）**	**p**
foot （フィート）	**ft**	**pounds per square inch （ポンド／平方インチ）**	**psi**
hertz （ヘルツ）	**Hz**	**radian （ラジアン）**	**rad**
hour （時）	**h**	**revolutions per minute （毎分回転数）**	**rpm**
inch （インチ）	**in**	**second (time) （秒）**	**s**
kilogram （キログラム）	**kg**	**volt （ボルト）**	**V**
kilometers per hour （キロメートル毎時）	**kph**	**watt （ワット）**	**W**
meter （メートル）	**m**		

主な取り扱い製品

開回路アキシャルピストンポンプ/モータ

ギヤポンプ/モータ

ファン駆動制御システム

閉回路アキシャルピストンポンプ/モータ

斜軸モータ

油圧トランスミッション

トラックミキサー用駆動装置

油圧トランスアクスル

統合システム

電子油圧制御装置

マイクロコントローラ/ソフトウェア

PLUS+1™ GUIDE

ディスプレイ

ジョイスティック/制御ハンドル

センサ

オービタルモータ

インバータ

電子油圧ステアリング

油圧パワーステアリング

油圧集積回路（HIC）

カートリッジバルブ

方向制御スプールバルブ

比例バルブ

11070058 · Rev FG · April 2009

ザウアーダンフォスグループの油圧システム
ー世界の油圧機器市場をリードする信頼のブランド

ザウアーダンフォスグループは、世界の車輌市場にシステムソリューションを提供する総合油圧機器メーカーです。

ザウアーダンフォスグループは、農業、建設、道路建設、物流、公共事業、林業、芝刈など、多くの市場に製品を提供しています。

ザウアーダンフォスグループは、お客様のニーズに最適なソリューションを提供し、お客様との密接なパートナーシップを通して新しい製品ならびにシステムを開発しております。

ザウアーダンフォスグループは、システムを構成するコンポーネントを一体化し、車輌設計者に最新のトータルシステムソリューションを提供しております。

ザウアーダンフォスグループは、世界各地の認定サービスセンターのネットワークを通じて、ワールドワイドなサービスを提供しております。

ザウアーダンフォス・ダイキン株式会社

アジア太平洋地域本部・日本地区本部・西日本営業所
〒651-0086 兵庫県神戸市中央区磯上通2-2-21 三宮グランドビル8F
本 部 代 表 TEL: 078-231-5001 FAX: 078-231-5004
西日本営業所 TEL: 078-231-5049 FAX: 078-231-5004

東日本営業所
〒101-0032 東京都千代田区岩本町2-5-12 サカエビル4F
TEL: 03-5835-0670 FAX: 03-3865-3270

サービス部・エンジニアリング部
〒566-0044 大阪府摂津市西一津屋1-1
サ ー ビ ス 部 TEL: 06-6349-7498 FAX: 06-6349-7401
エンジニアリング部 TEL: 06-6349-7562 FAX: 06-6349-7401

www.sauer-danfoss-daikin.com

仕様など記載内容は予告なく変更する場合がありますのでご了承ください。
このカタログに記載された内容の無断転載は固くお断りいたします。