

SIEMENS

SIMATIC

S7-1500

入門

ようこそ

自動化タスク

1

ハードウェアセクション

2

ソフトウェアセクション

3

セキュリティ

4

法律上の注意

警告事項

本書には、ユーザーの安全を確保し製品の損傷を防止する上で守るべき注意が記載されています。ユーザーの安全に関する注意事項は、安全警告サインで強調表示されています。このサインは、物的損害に関する注意事項には表示されません。注意事項は、危険度によって以下の等級に分類されています。

⚠ 危険
回避しなければ、直接的な死亡、重傷に 至る 危険な状態を示します。

⚠ 警告
回避しなければ、死亡、重傷に 至るおそれのある 危険な状況を示します。

⚠ 注意
回避しなければ、軽度または中度の人的傷害を引き起こす おそれのある 危険な状況を示します(安全警告サイン付き)。

注記
回避しなければ、物的損害を引き起こす おそれのある 危険な状況を示します(安全警告サインなし)。

複数の危険レベルに相当する場合、通常は最も危険度の高い(番号の低い)事項が表示されることになっています。安全警告サイン付きの人的傷害に関する注意事項があれば、物的損傷に関する警告が付加されます。

有資格者

装置/システムのセットアップおよび使用にあたっては必ず本書を参照してください。機器の据え付けおよび操作は**有資格者**のみが行うものとします。有資格者とは、法的な安全規制/規格に準拠して、接地の取り付け、電気回路、設備およびシステムの設定に携わることを承認されている技術者のことをいいます。

シーメンス製品を正しくお使いいただくために

以下の事項に注意してください。

⚠ 警告
シーメンス製品は、カタログおよび付属の技術説明書の指示に従ってお使いください。他社の製品または部品との併用は、弊社の推奨もしくは許可がある場合に限りです。シーメンス製品を正しく安全にお使いいただくには、適切な運搬、保管、組み立て、据え付け、配線、始動、操作、保守を行ってください。ご使用になる場合は、許可された範囲を必ず守ってください。付属の技術説明書に記述されている指示を遵守してください。

商標

本書において®で識別されるすべての名称は、Siemens AG の登録商標です。本書に記載するその他の称号は商標であり、第三者が自己の目的において使用した場合、所有者の権利を侵害することになります。

免責事項

本書のハードウェアおよびソフトウェアに関する記述と、実際の製品内容との一致については検証済みです。しかしながら、本書の記述が実際の製品内容と異なる可能性もあり、完全な一致が保証されているわけではありません。記載内容については定期的に検証し、訂正がある場合は次の版で更新いたします。

ようこそ

ようこそ

「TIA ポータル V13」 入門によるこそ

本入門書では、「カラーミキシングプラント」用の自動制御ソリューションを作成するために、TIA ポータルを使った CPU SIMATIC S7-1500 の使用方法を紹介します。ビデオクリップは、自動化タスク用のソリューションを作成する方法を紹介します。

最初の部分では、ハードウェアを組み立て、設定用 PC を準備します。

2 番目の部分では、カラーミキシングプラントの例を使用して、CPU と HMI ビジュアライゼーションを構成定義します。

さらに、自動制御ソリューションのオプションと拡張部分について説明します。

セキュリティ情報

シーメンスは、弊社の製品およびソリューションに対して、プラント、ソリューション、機械、設備またはネットワークの安全な運転をサポートする産業セキュリティファンクションを提供します。これらの製品は、産業セキュリティコンセプト全体にとって重要な構成要素となります。この点を踏まえて、シーメンスの製品は日々発展を続けています。そのため、弊社の製品に関する最新情報を常に確認することを強くお勧めします。

シーメンス製品とソリューションのセキュアなオペレーションのために、適切な予防措置 (セル保護コンセプトなど) を取ることで、各コンポーネントを包括的な、最先端の産業セキュリティコンセプトに統合する必要があります。使用されている可能性があるサードパーティ製品についても同様に考慮する必要があります。産業セキュリティの詳細については、インターネットを参照してください (<http://www.siemens.com/industrialsecurity>)。

常に弊社製品の最新情報を入手するには、製品情報のニュースレターにご登録ください。産業セキュリティの詳細については、インターネットを参照してください (<http://support.automation.siemens.com>)。

目次

	ようこそ	3
1	自動化タスク	7
1.1	概要	7
1.2	サンプルプロジェクト	8
2	ハードウェアセクション	15
2.1	概要	15
2.1.1	必要条件	15
2.1.2	追加情報	16
2.2	組み立て部品の取り付け	17
2.2.1	概要	17
2.2.2	組み立て部品の取り付け	17
2.3	配線	20
2.3.1	概要	20
2.3.2	配線ルール	21
2.3.3	主電源接続プラグの配線	22
2.3.4	電源モジュール(PM)の CPU への配線	24
2.3.5	電位ブリッジ回路	25
2.3.6	デジタル入力モジュールの配線	25
2.3.7	デジタル出力モジュールの配線	27
2.3.8	フロントコネクタの配線	29
2.4	電源オン	31
2.4.1	概要	31
2.4.2	電源オン	31
2.4.3	ディスプレイで IP アドレスを割り当て	33
3	ソフトウェアセクション	35
3.1	プロジェクトとハードウェアの作成	35
3.1.1	TIA ポータルの紹介	35
3.1.2	プロジェクトの作成	37
3.1.3	S7-1500 CPU の作成	39
3.1.4	ハードウェア検出の実行	41
3.1.5	ET 200 インターフェースモジュールの作成	42
3.1.6	ET 200 インターフェースモジュールのネットワーク接続	43
3.1.7	ET 200SP 用の入出力モジュールとサーバーモジュールの作成	45
3.1.8	ET 200MP 用の入出力モジュールの作成	47
3.1.9	ET 200 に名前を割り当てる	48

3.2	プログラムの作成	49
3.2.1	ブロックライブラリのロード	49
3.2.2	プログラムブロック Main [OB1]の削除	51
3.2.3	プログラムブロックのコピー	52
3.2.4	周期割り込み OB	53
3.2.4.1	周期割り込み OB – サイクルタイムと位相	53
3.2.4.2	サイクルタイムの変更	54
3.2.5	タグテーブルのコピー	55
3.2.6	プロジェクトのコンパイル	56
3.2.7	プロジェクトを CPU にダウンロード	58
3.2.8	最適化ブロックアクセス	60
3.2.8.1	概要	60
3.2.8.2	最適化[Filling]データブロックの拡張と再ロード	61
3.2.9	ブロックのバージョン管理	66
3.2.10	保持設定	69
3.2.11	EN/ENO メカニズムの有効化	72
3.2.12	コメントファクションの使用	74
3.2.13	ローカルエラーハンドリング	75
3.2.13.1	ブロック内のエラー処理	75
3.2.13.2	ローカルエラーハンドリング用のブロックのダウンロード	77
3.2.13.3	ローカルエラーハンドリングなしでエラーを生成	79
3.2.13.4	ローカルエラーハンドリングを使用してエラーを生成	80
3.3	ビジュアルライゼーションの設定	82
3.3.1	現在のサンプルプロジェクト	82
3.3.2	HMI 構成	82
3.3.2.1	概要	82
3.3.2.2	SIMATIC HMI コンフォートパネル	83
3.3.2.3	HMI 画面	84
3.3.2.4	追加のコントロールエレメント	85
3.3.2.5	レシピ	86
3.3.2.6	アーカイブ	87
3.3.2.7	ユーザー定義ファンクション	88
3.3.2.8	ユーザー管理	89
3.3.2.9	多言語の使用	90
3.3.2.10	レポート	92
3.3.3	ライブラリからの HMI デバイスの挿入	94
3.3.3.1	オブジェクトをライブラリに保存	94
3.3.4	HMI 接続の構成定義	95
3.3.4.1	デバイス間の通信(RT Professional、RT Advanced、ベーシックパネル、パネル、 コンフォートパネル、RT Unified).....	95
3.3.4.2	HMI 接続の構成定義	96
3.3.4.3	HMI タグの接続	98
3.3.5	システム診断の構成定義	100
3.3.5.1	システム診断の基本(RT Advanced、パネル、コンフォートパネル)	100
3.3.5.2	システム診断ビュー(RT Advanced、パネル、コンフォートパネル)	101
3.3.5.3	システム診断ビューの構成定義(RT Advanced、ベーシックパネル、パネル、コンフォー トパネル)	104
3.3.6	HMI デバイスのシミュレーション	106
3.3.6.1	シミュレーションの基本(RT Professional、RT Advanced、パネル、コンフォートパ ネル).....	106
3.3.6.2	シミュレーションでパネルの操作	107

3.4	プロジェクトをプログラミングツールにアップロード	111
3.4.1	CPU をプロジェクトにアップロード	111
3.5	Inter Project Engineering による Team Engineering	113
3.5.1	「Inter Project Engineering」の基本	113
3.5.2	IPE ファイルの作成	114
3.5.3	IPE ファイルのインポート	115
4	セキュリティ	119
4.1	CPU の保護ファンクションの概要	119
4.2	ディスプレイを使用して、追加のアクセス保護を設定	120
4.3	ノウハウプロテクト	121
4.4	コピー保護の設定	124
4.5	CPU ロックによる保護	125
4.6	CPU のアクセス保護の設定	125
4.7	HMI 接続の保護の設定	128
	索引	129

自動化タスク

1.1 概要

概要

次のセクションでは、自動化タスクについて説明します。

アプリケーション例、ハードウェアコンフィグレーション、サンプルプロジェクトのコンポーネントについて詳しく紹介します。

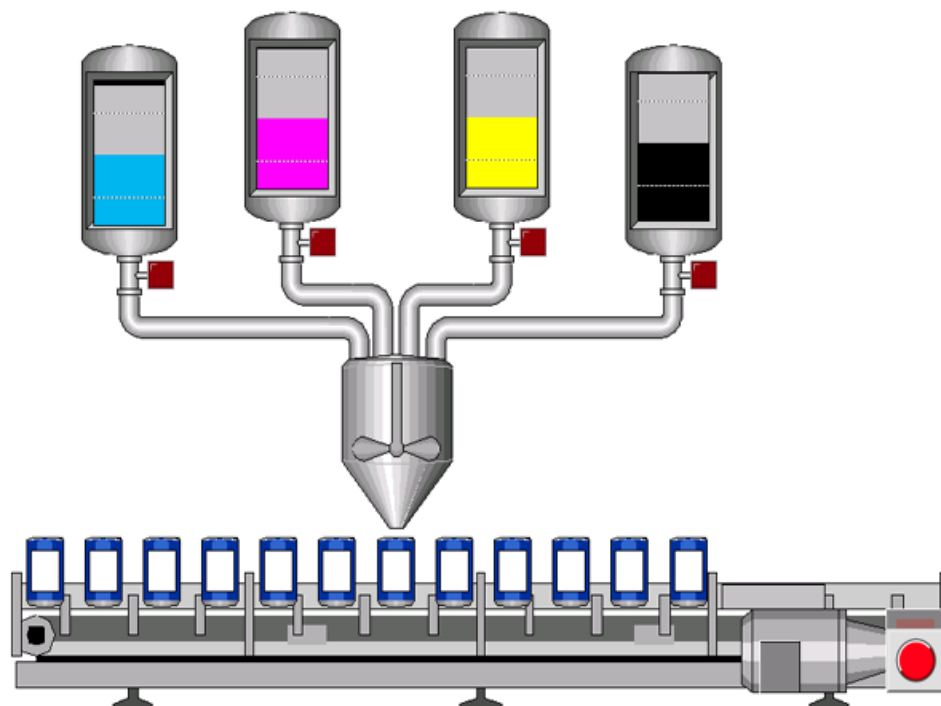
アプリケーション例

本入門書のアプリケーション例として、事前に選択したカラーレシピを混合し、充填するカラーミキシングプラントを取り上げます。

レシピには4つの色成分、シアン、マゼンタ、黄色、黒色があり、これはCMYK色空間の色です。

充填は以下の4つの手順で行われます。

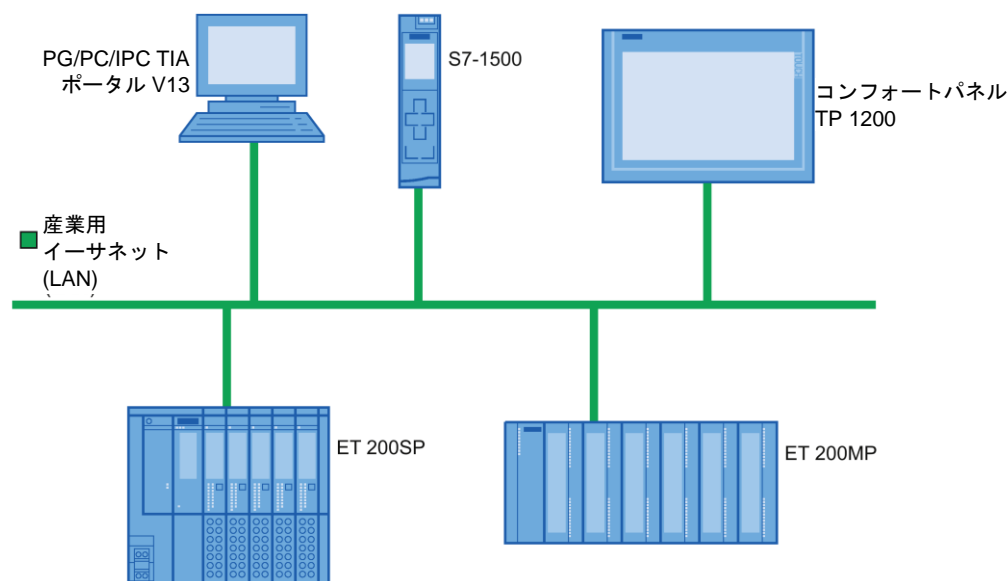
- HMI レシピファンクションを使用して、混色の選択
- それぞれのタンクのバルブを開けて、レシピ成分または4つの基本色を充填
- 色のミキシング
- 完成した混色を缶に充填し、コンベヤベルトで搬送



ハードウェアコンフィグレーションの設計

ハードウェアコンフィグレーションは、次のデバイスから構成されます。

- CPU 1511-1 PN (S7-1500 電源モジュール、デジタル入力モジュール、およびデジタル出力モジュール付き)
- TIA ポータルでシミュレート可能な HMI パネル TP1200 コンフォート
- リモート ET 200MP I/O システム(IM 155-5 PN ST インターフェースモジュール、およびデジタル入出力モジュール付き)
- リモート ET 200SP I/O システム(IM 155-6 PN ST インターフェースモジュール、デジタル入力モジュール、デジタル出力モジュール、およびサーバーモジュール付き)



1.2 サンプルプロジェクト

アプリケーション用のサンプルプロジェクト

TIA ポータルを使用してカラーミキシングシステムを構成するには、サンプルプロジェクト「Color_Filling_Station」を作成します。

次のプロジェクトコンポーネントは、既にサンプルプロジェクト用に存在します。

- CPU のプログラムブロック
- コンフォートパネルでの HMI のビジュアルイゼーション

このセクションでは、サンプルプロジェクトの個々のプロジェクトコンポーネント間の関係を説明します。

後で、必要なコンフィグレーション手順を自分で実際に行うことになります。

HMI

CPU

①

Color selection recipe

Recipe Name:

No.:

Data Record Name:

No.:

Entry Name

Value

Status bar

色選択レシピ

②

Elements

Data records

	Name	Display name	Tag
	Cyan	Recipe_Cyan	HMI_Recipe_Curr...
	Magenta	Recipe_Magenta	HMI_Recipe_Curr...
	Yellow	Recipe_Yellow	HMI_Recipe_Curr...
	Black	Recipe_Black	HMI_Recipe_Curr...

色要素

④

Amount

Number of cans

10

9

8

7

6

5

4

3

2

1

0

5

スライダー

③

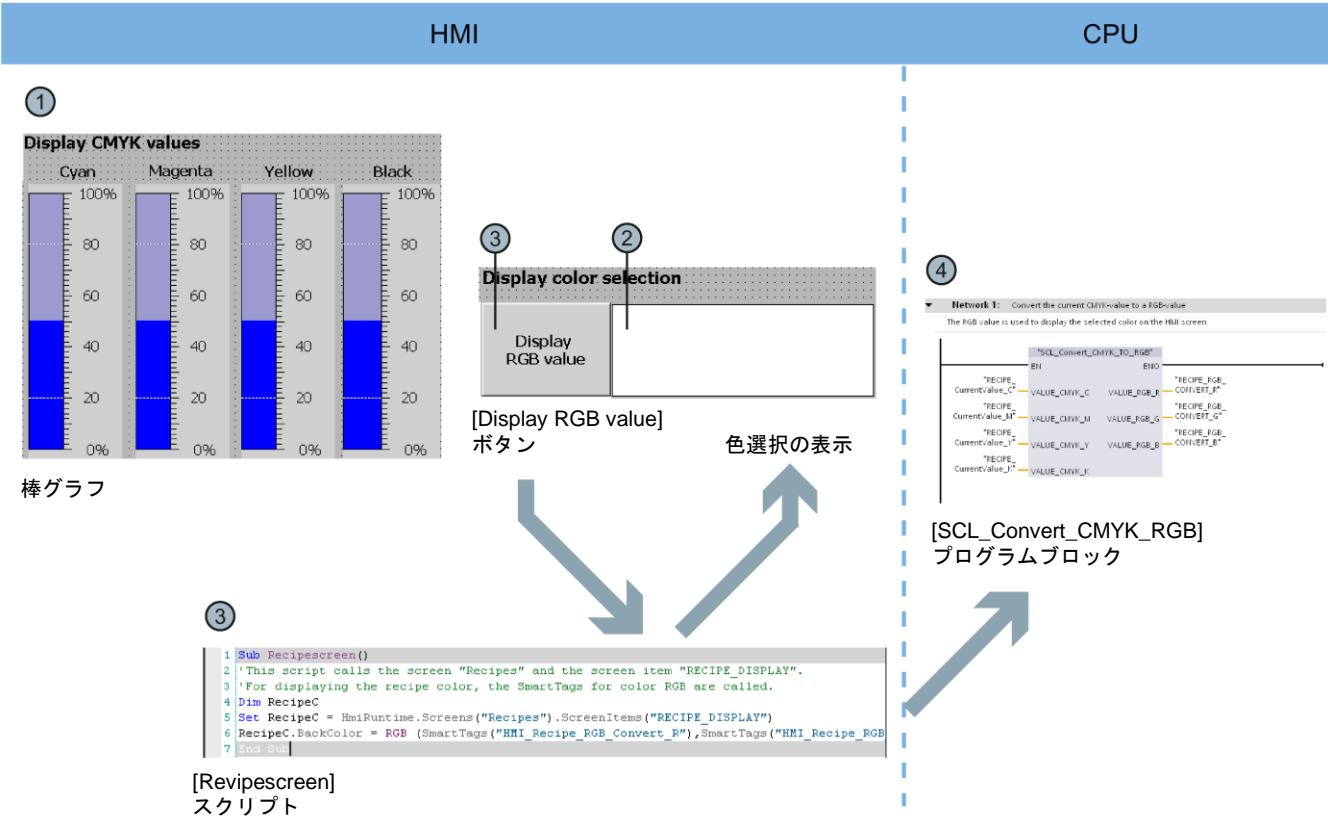
Tags_Filling_Process

	Name	Data type
1	START_FILL_RECIPE	Bool
2	FILLING_VALVE_OPEN	Bool
3	VALVE_CONVEYOR_BUSY	Bool
4	START_CONVEYOR	Bool
5	FILLING_DONE	Bool
6	FILLING_CONVEYOR_ET	Time
7	FILLING_VALVE_ET	Time
8	FILLING_COUNTER	Int
9	STOP_RESET_FILLING_PROCESS	Bool
10	RECIPE_CurrentValue_C	DInt
11	RECIPE_CurrentValue_M	DInt
12	RECIPE_CurrentValue_Y	DInt
13	RECIPE_CurrentValue_K	DInt
14	RESET_FILLING_Level_C	Bool
15	RESET_FILLING_Level_M	Bool
16	RESET_FILLING_Level_Y	Bool
17	RESET_FILLING_Level_K	Bool
18	ALERT_TANKS	Bool

PLC タグ

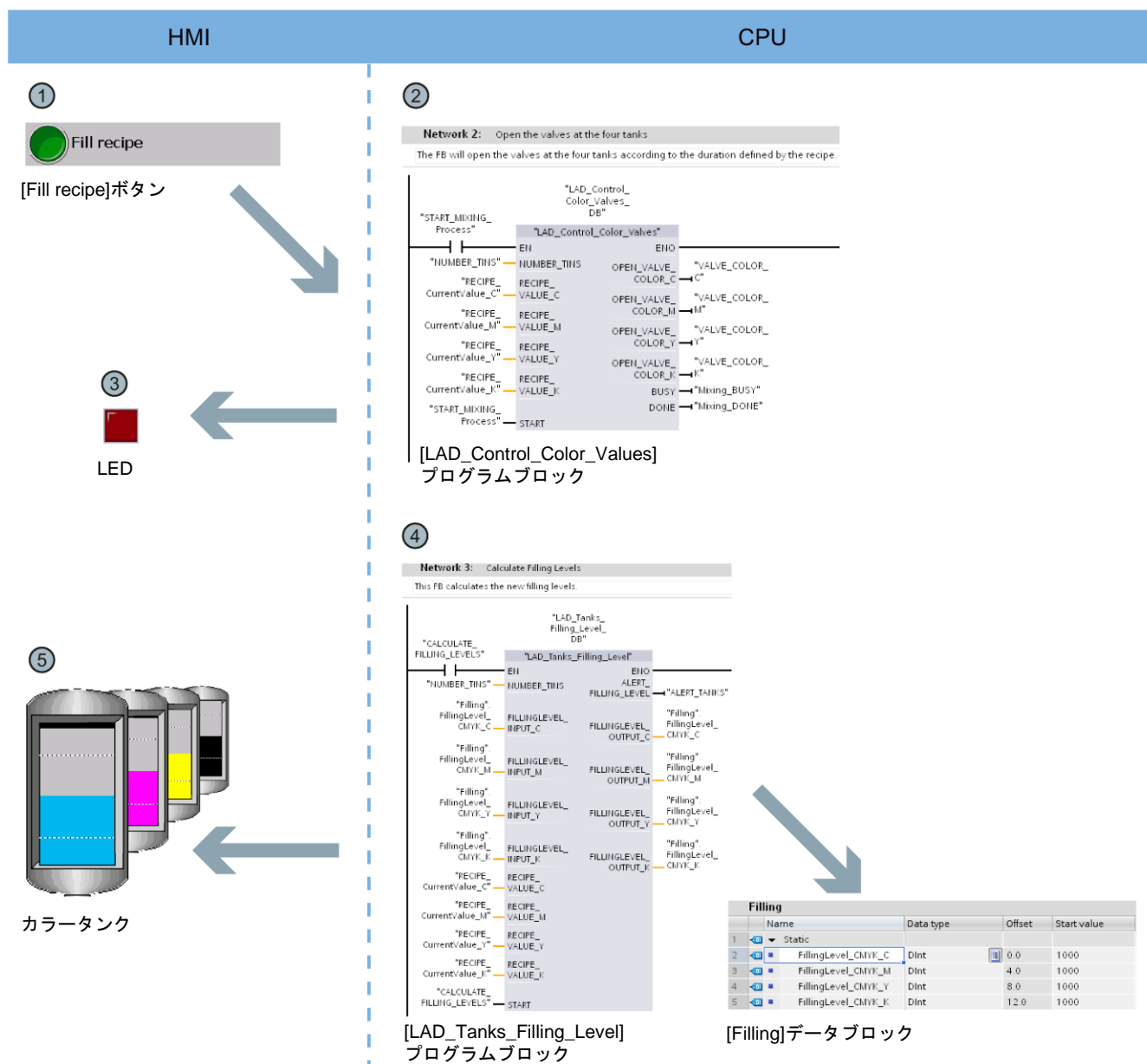
- ① HMI 画面[Recipes]には、[Color selection recipe]があります。これは、TIA ポータルのライブラリの作成済みのオブジェクトです。このオブジェクトを使用して、データレコードを選択し、新しいデータレコードを作成できます。
- ② データレコード(混色)とエレメント(色成分)は、HMI エディタの[Recipes]に保存されます。各々の混色は、色成分シアン(C)、マゼンタ(M)、黄色(Y)、黒色(K)から構成されます。混色での 4 つの色成分のそれぞれの割合は、[Recipes]エディタに保存されます。
- ③ それぞれの色成分の値は、混色をロードするときに PLC タグに書き込まれます。PLC タグは、CPU の[Tags_Filling_Process]タグテーブルに保存されます。
- ④ HMI 画面[Recipes]にもスライダーがあります。このスライダーを使用して、充填する必要のある年数を指定します。

CMYK と RGB 値の表示



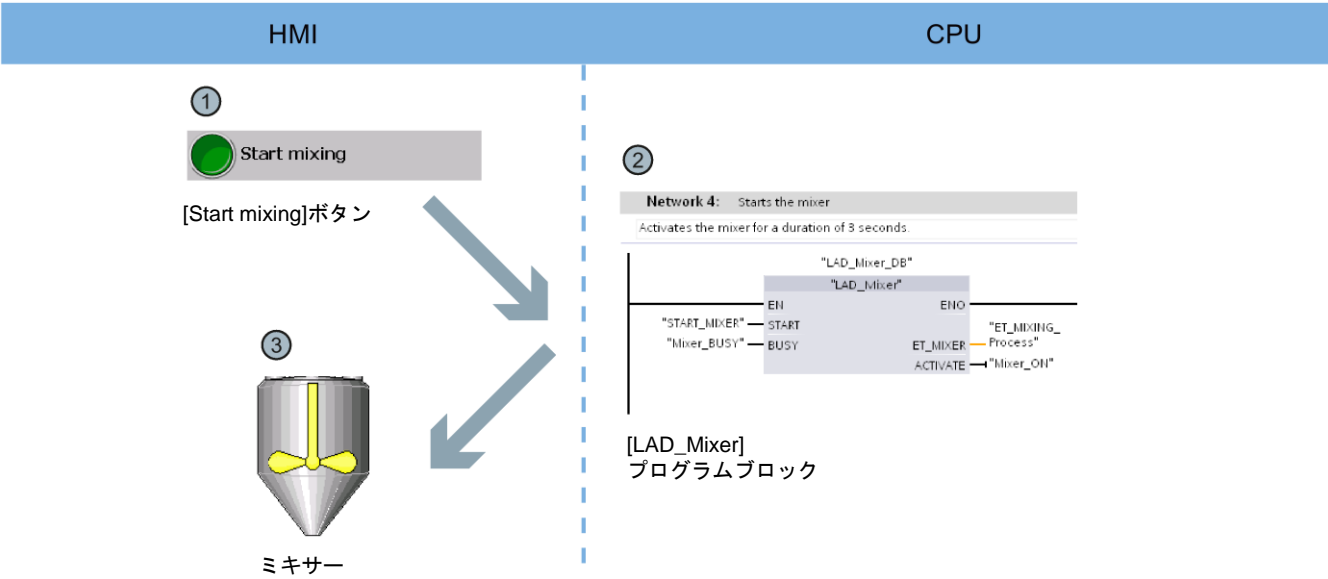
- ① 必要な混色を HMI 画面[Recipes]で選択すると、CMYK 色空間での値が棒グラフで表示されます。
- ② 混色は、他の表示形式で示すこともできます。これには、[Recipescreen]スクリプトを実行する必要があります。
- ③ [Recipescreen]スクリプトを実行するには、[Display RGB Value]ボタンをクリックします。CMYK 値は直接画面に出力できないので、スクリプトは CMYK 値に割り当てられた RGB 値をディスプレイに割り当てます。
- ④ 必要な RGB 値は、[SCL_Convert_CMYK_TO_RGB]プログラムブロックによって計算されます。

レシピの充填



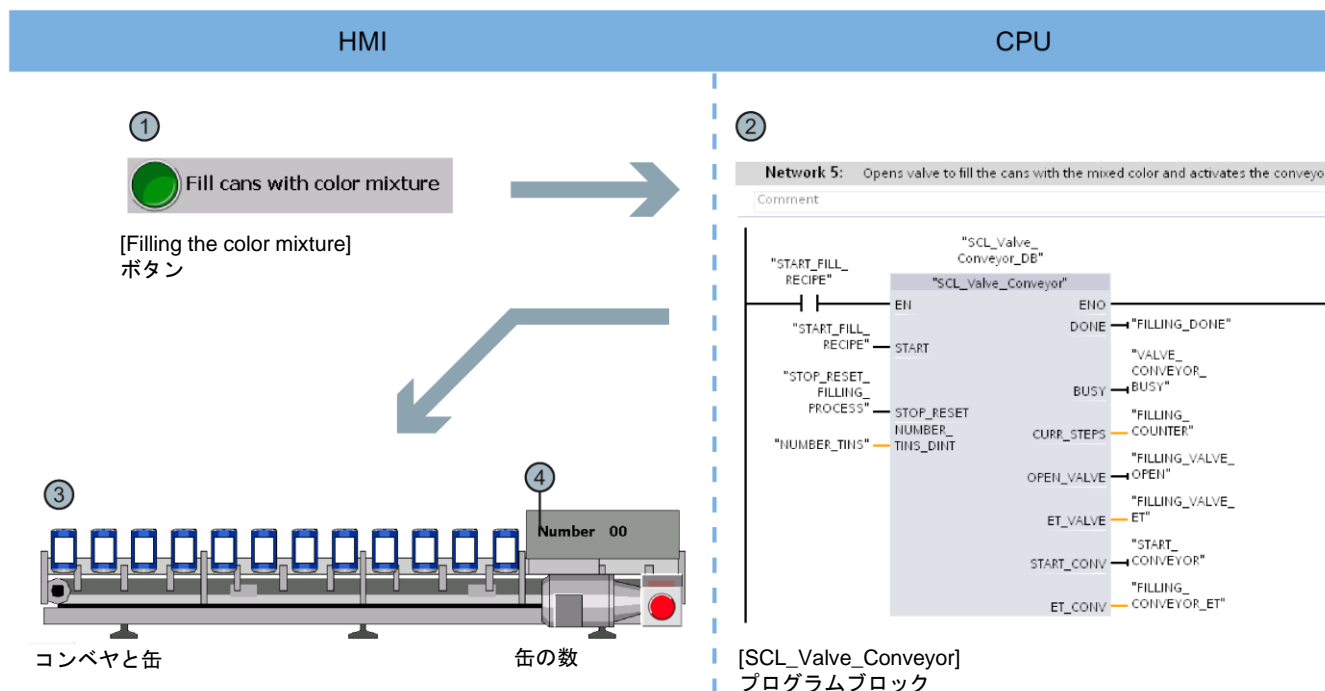
- ① [Fill recipe]ボタンは、HMI 画面[Start screen]の色成分の充填を開始します。このボタンは、[LAD_Control_Color_Valves]プログラムブロックを有効にします。
- ② プログラムブロックは、指定したレシピと充填が必要な缶数に基づいて混色のために4つのバルブをどのくらいの時間開く必要があるか計算します。
- ③ タンクの下 LED は、バルブが開いていることを示します。
- ④ [LAD_Tanks_Filling_Level]プログラムブロックは、充填と同時に実行されます。プログラムブロックは、タンクの占有レベルのためにタンクに残存している量を計算します。タンクの占有レベルは、グローバルデータブロック[Filling]に保存されます。
- ⑤ HMI 画面の占有レベルインジケータは、直接グローバルデータブロックとリンクし、各ランタイム取得間隔で更新されます。

ミキシングプロセスの開始



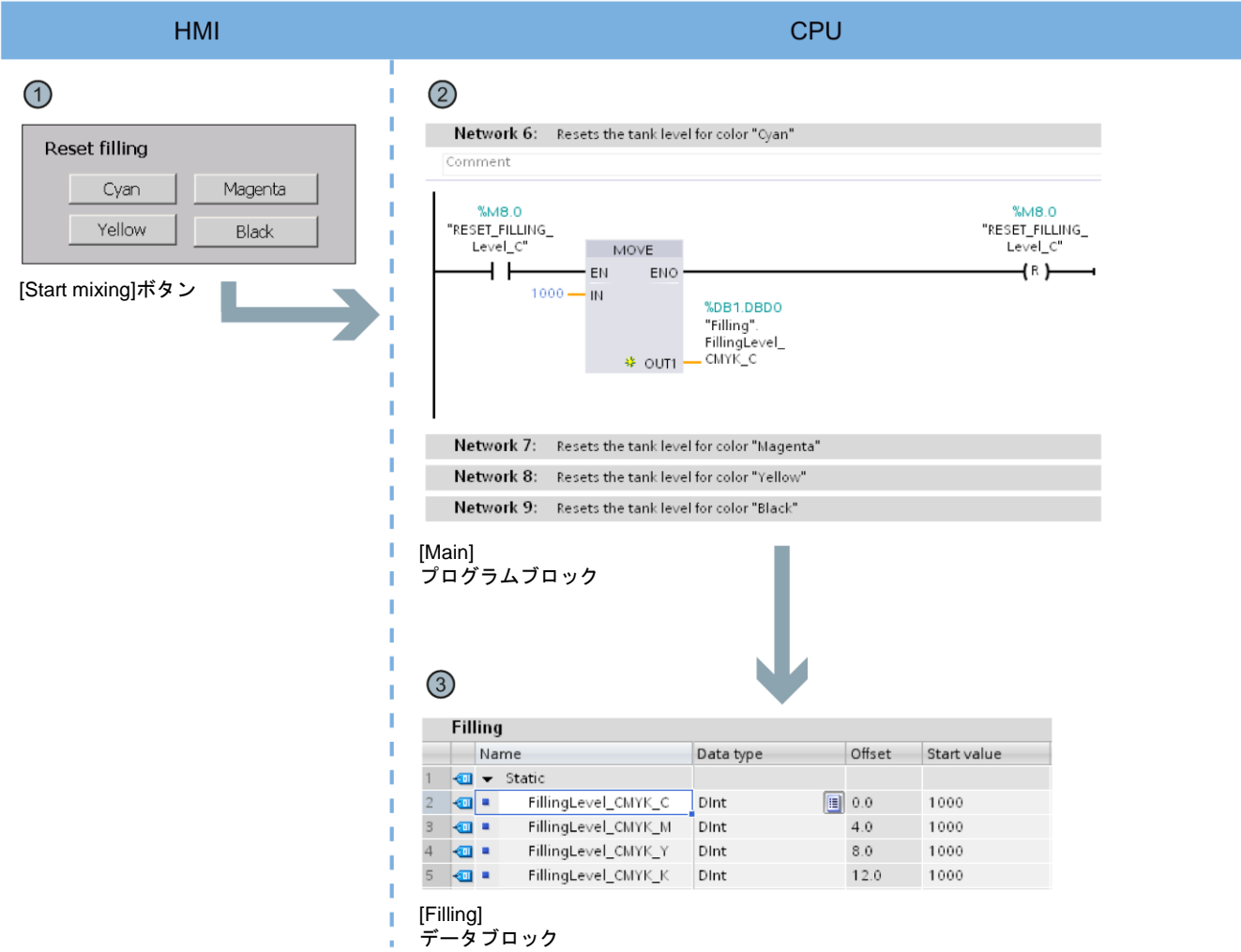
- ① [Start mixing process]ボタンは、HMI 画面[Start screen]のカラーミキシングプラントのミキサーを起動します。
- ② [LAD_Mixer]プログラムブロックは、CPU 側でこのために呼び出されます。3 秒間、ミキサーを稼働します。
- ③ ミキサーの稼働は、HMI 画面の点滅で示されます。

混色の充填



- ① [Fill Color Mixture]ボタンは、HMI 画面[Start screen]で缶の充填を開始します。
- ② [SCL_Valve_Conveyor]プログラムブロックは、CPU 側でこのために有効になります。バルブとコンベヤベルトを制御します。
- ③ 缶は、HMI 画面のコンベヤベルトの動きに従ってアニメーション表示されます。
- ④ カウンタは、既に充填された缶数を示します。

占有レベルのリセット



- ① HMI 画面[Start screen]には、4 つの色タンクのそれぞれの占有レベルをリセットするボタンがあります。
- ② それぞれの占有レベルのリセット機能は、ネットワーク 6 から 9 の[Main]プログラムブロックに実装されています。
- ③ ネットワーク 6 から 9 は、グローバルデータブロック[Filling]の値を開始値にリセットします。

ハードウェアセクション

2.1 概要

Totally Integrated Automation Portal (TIA ポータル)を使用する新しい SIMATIC S7-1500 コントローラファミリーはさまざまな新しいオプションを提供し、機械の生産性を向上し、エンジニアリングプロセスをさらにいっそう効率化します。本入門書でオプションを紹介します。

最初の基本ステップで、新しいハードウェアをさらに詳しく知ることができるようになります。また、SIMATIC STEP 7 V13 (TIA ポータル)を使用して SIMATIC S7-1500 を構成しプログラムする方法を紹介します。SIMATIC HMI コンフォートパネルと SIMATIC WinCC Advanced V13 (TIA ポータル)の接続、または SIMATIC WinCC Professional V13 (TIA ポータル)との接続で、基本手順が完了します。

2.1.1 必要条件

ハードウェア要件

本入門書のハードウェアセクションを実装するには、以下のハードウェアが必要です。

- 1 × CPU 1511-1 PN (6ES7511-1AK00-0AB0)
- 1 × S7-1500 電源モジュール PM 70W 120/230VAC (6EP1332-4BA00)
- 1 × 取り付けレール(6ES7590-1AB60-0AA0)
- 1 × デジタル入力モジュール DI 16x24VDC SRC BA (6ES7521-1BH50-0AA0)
- 1 × デジタル出力モジュール DQ 16x24VDC/0.5A ST (6ES7522-1BH00-0AB0)
- 2 × フロントコネクタ(6ES7592-1AM00-0XB0)
- 1 × SIMATIC メモリカード、4 MB 搭載(6ES7954-8LBxx-0AA0 など)
- 1 × イーサネットケーブル

また、上述のハードウェアは以下のスタータパッケージの一部です。

スタータパッケージ S7-1500、ソフトウェア付き: 6ES7511-1AK00-4YB5

ソフトウェア要件

本入門書のソフトウェアセクションを実装するには、以下のソフトウェアが必要です。

- SIMATIC STEP 7 Professional V13
- SIMATIC WinCC Advanced V13 または SIMATIC WinCC Professional V13



警告

重大な人的傷害に至るおそれのある状況

プラントやシステムの S7-1500 オートメーションシステムは、アプリケーションの関連分野に基づいた特別な規格や規制によって管理されます。IEC 60204-1(一般的な機械安全要件)のような適用される安全法規や事故防止規則を順守してください。

これらの規則に従わなければ、重傷または物的損害を引き起こすおそれがあります。

2.1.2 追加情報

使用されるハードウェアの詳細な情報は、以下のリンクから利用できます。

- CPU 1511-1 PN (6ES7511-1AK00-0AB0)
(<http://support.automation.siemens.com/WW/view/en/68020492>)
- S7-1500 電源モジュール PM 70W 120/230VAC (6EP1332-4BA00)
(<http://support.automation.siemens.com/WW/view/en/68036174>)
- DI 16x24 V DC SRC BA デジタル入力モジュール(6ES7521-1BH50-0AA0)
(<http://support.automation.siemens.com/WW/view/en/59191844/>)
- DQ 16x24 V DC/0.5A ST デジタル出力モジュール (6ES7522-1BH00-0AB0)
(<http://support.automation.siemens.com/WW/view/en/59193401>)

2.2 組み立て部品の取り付け

2.2.1 概要

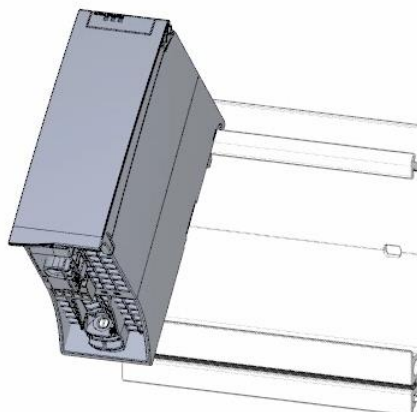
組み立て部品の取り付け

このセクションで、構造部分の取り付けをします。

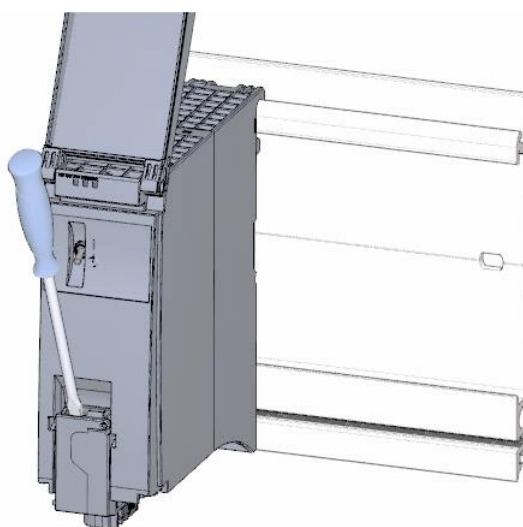
2.2.2 組み立て部品の取り付け

手順

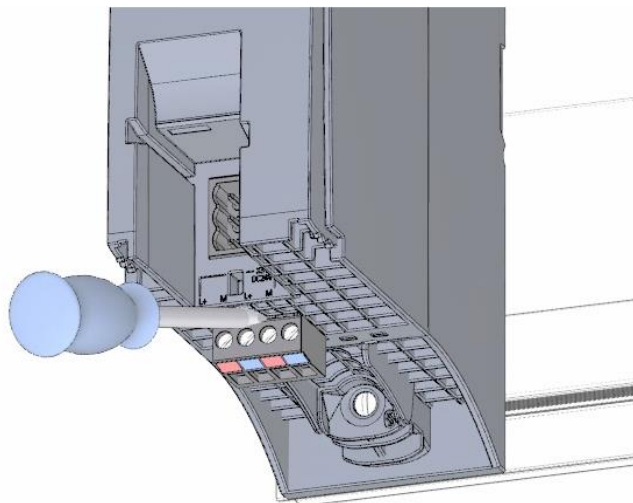
1. 電源モジュール(PM)を取り付けレールに取り付けます。



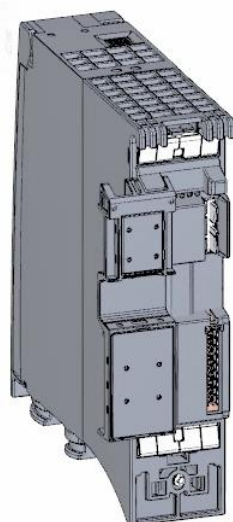
2. フロントカバーを開き、主電源の接続プラグを引き出します。



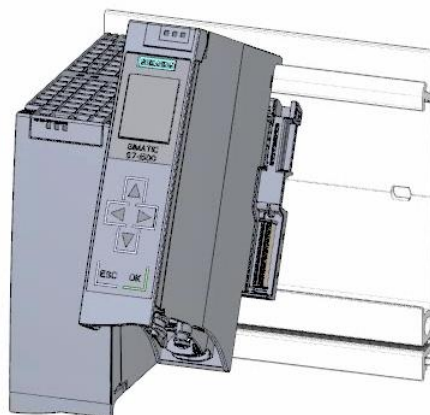
3. 4 極接続プラグを取り外し、電源モジュール(PM)をしっかりとネジ止めします。



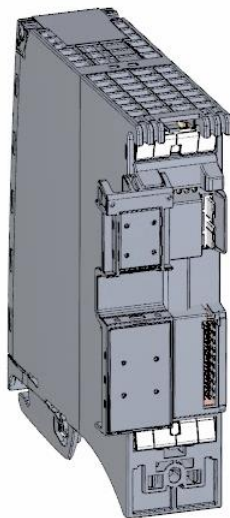
4. U コネクタを CPU の背面に挿入します。



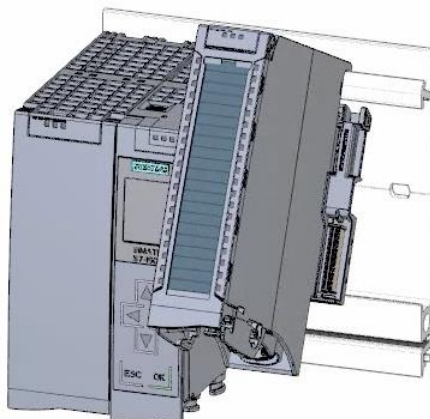
5. CPU を取り付けレールに取り付け、しっかりとネジ止めします。



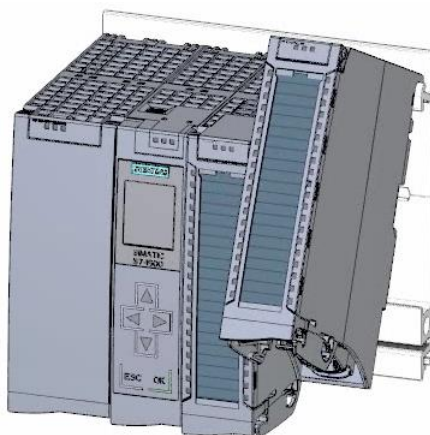
6. Uコネクタをデジタル入力モジュールの背面に挿入します。



7. デジタル入力モジュールを取り付けレールに装着し、しっかりとネジ止めします。



8. デジタル出力モジュールを取り付けレールに装着し、しっかりとネジ止めします。



結果

組み立て部品が取り付けられました。

2.3 配線

2.3.1 概要

組み立て部品の配線

このセクションで組み立て部品を取り付けます。

 危険
電源モジュール用の主電源ケーブルは、配線作業の間に電源に接続してはなりません。

2.3.2 配線ルール

プラントやシステム S7-1500 CPU のオペレーションは、関連する応用分野に基づいた特別な一連のルールや規則によって定義されます。

S7-1500 システムの説明に S7-1500 の操作のための一般的なルールや規則が記載されています(<http://support.automation.siemens.com/WW/view/en/59191792>)。

CPU 用の配線ルール

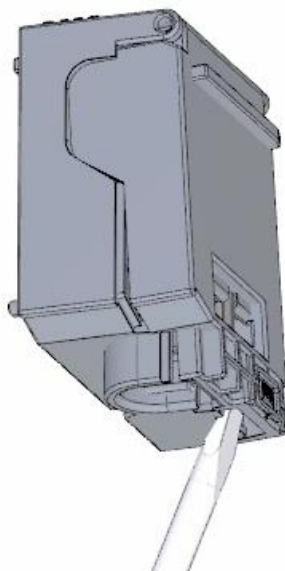
配線ルール...		CPU	40 ピンフロントコネクタ (ネジ型接続)	負荷電源
ソリッドワイヤ用に接続可能なワイヤ断面		—	最大 0.25 mm ²	—
		—	AWG*: 24	—
より線用の接続可能なワイヤ断面	ワイヤ端部のフェルールなし	0.25~2.5 mm ²	0.25~1.5 mm ²	1.5 mm ²
		AWG*: 24~16	AWG*: 24~16	AWG*: 16
	ワイヤ端部のフェルール付き	0.25~2.5 mm ²	0.25~1.5 mm ²	1.5 mm ²
		AWG*: 24~16	AWG*: 24~16	AWG*: 16
接続あたりのワイヤの数		1	1 本または 2 本のケーブルの組み合わせで、最大 1.5 mm ² (合計)、同じワイヤ端部のフェルール付き	1
ストリップされた配線の長さ		10~11 mm	10~11 mm	7~8 mm
DIN 46228 に準拠したエンドスリーブ	プラスチックスリーブなし	設計 A、10 mm 長	設計 A、10 mm および 12 mm 長	設計 A、7 mm 長
	プラスチックスリーブ付き、0.25~1.5 mm ²	設計 E、10mm 長	設計 E、10 mm および 12 mm 長	設計 A、7 mm 長
シース直径		—	—	8.5 mm
工具		3~3.5 mm フィリップス(プラス)ドライバー、円錐デザイン	3~3.5 mm フィリップス(プラス)ドライバー、円錐デザイン	3~3.5 mm フィリップス(プラス)ドライバー、円錐デザイン
接続方法		プッシュイン端子	ネジ端子	ネジ端子
締め付けトルク		—	0.4 Nm~0.7 Nm	0.5 Nm~0.6 Nm

* AWG: 米国ワイヤゲージ規格

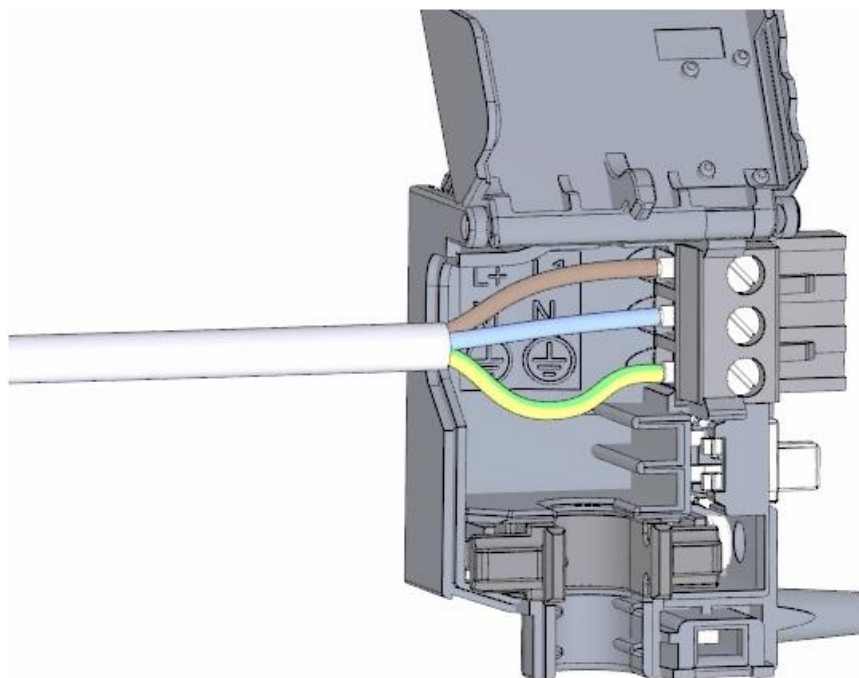
2.3.3 主電源接続プラグの配線

手順

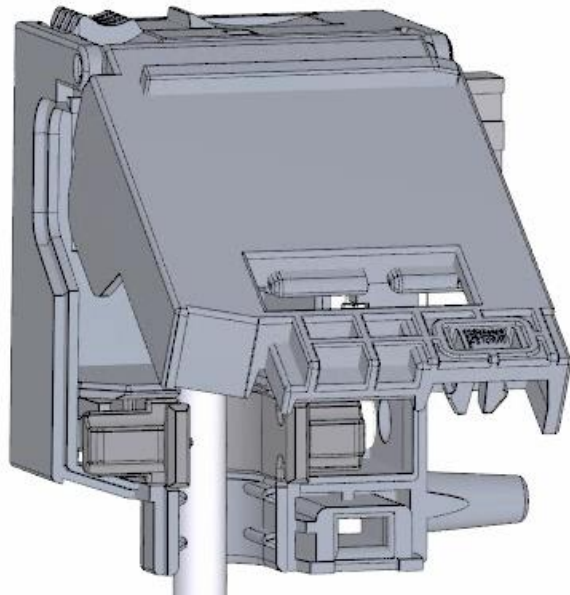
1. 適切な工具を使用してコネクタカバーをこじ開けます。



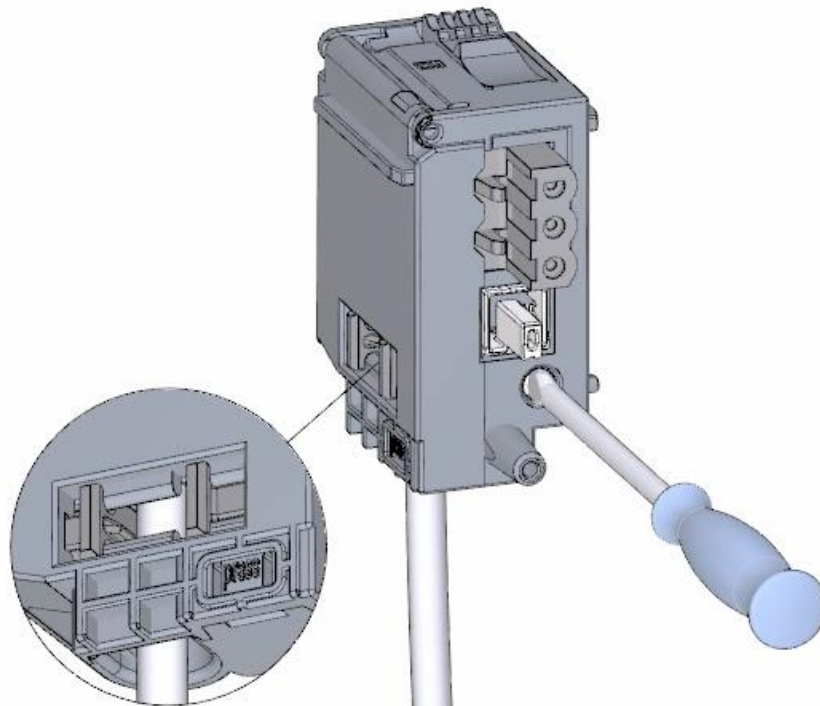
2. 接続図に従って主電源ケーブルをプラグに接続します。プラグの片側に、プラグが認定されている電圧についての情報が記載されています。プラグ背面に適切にコードエレメントを挿入して電圧を選択します。



3. カバーを閉じます。



4. 主電源の接続プラグのフロントカバーのネジをしっかりと締めます。



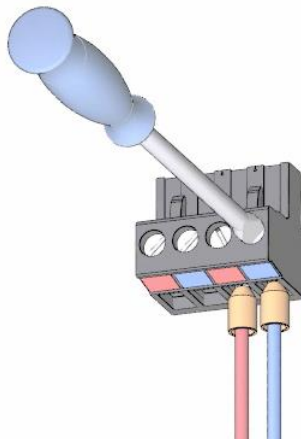
結果

主電源の接続プラグがこれで配線されます。

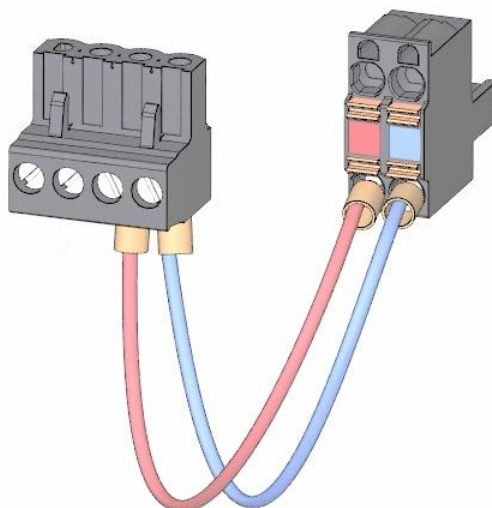
2.3.4 電源モジュール(PM)の CPU への配線

手順

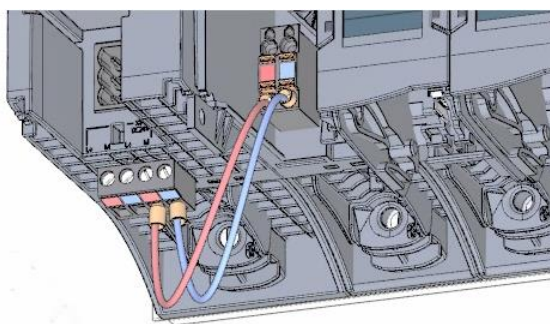
1. 電源モジュール(PM)の 4 ピンコネクタプラグを配線します。



2. 4 ピンコネクタプラグと CPU の 4 ピン主電源接続プラグを配線します。



3. 電源モジュール(PM)を CPU に接続します。



結果

電源モジュールがこれで CPU に配線されます。

2.3.5 電位ブリッジ回路

電位ブリッジ回路の用途

負荷グループに同じ電位(非絶縁)を供給したい場合、フロントコネクタ用に支給された電位回路ブリッジを使用します。これは、2本のワイヤでクランプユニットに配線するのを避けられることを意味します。

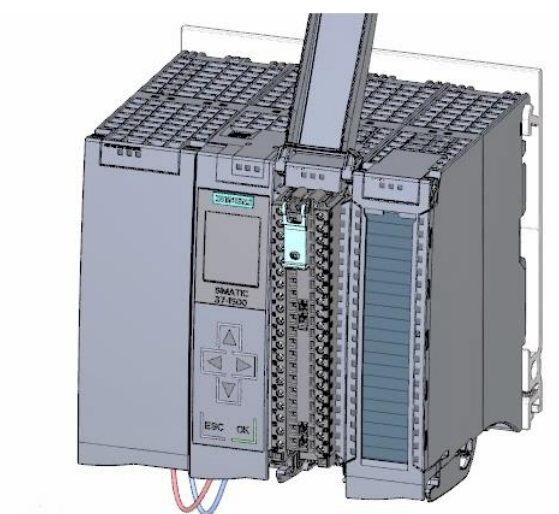
ヒント

フロントコネクタに端子 40 (M)および 39 (L+)を使用して、電位を次のモジュールにループします。

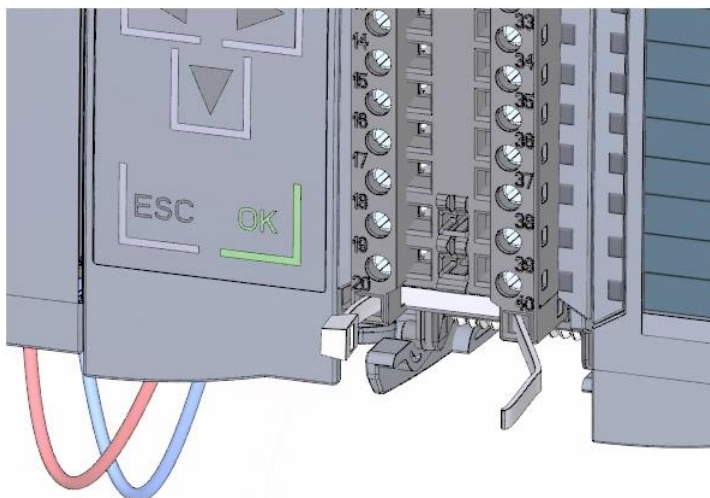
2.3.6 デジタル入力モジュールの配線

手順

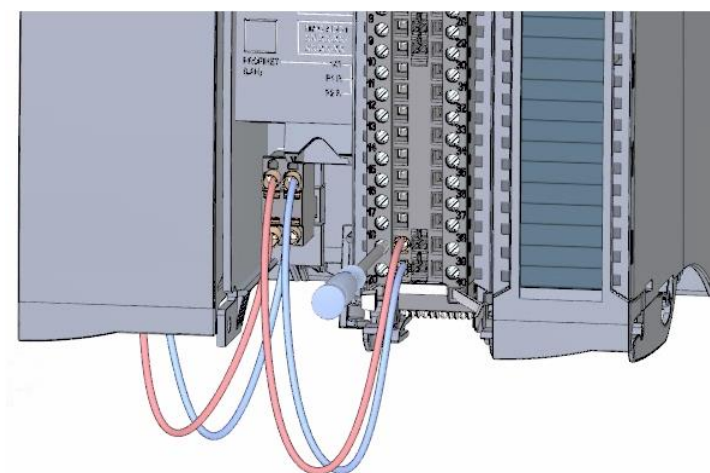
1. フロントコネクタを事前配線位置に挿入します。事前配線位置では、フロントコネクタとモジュールの間に電氣的接続はありません。



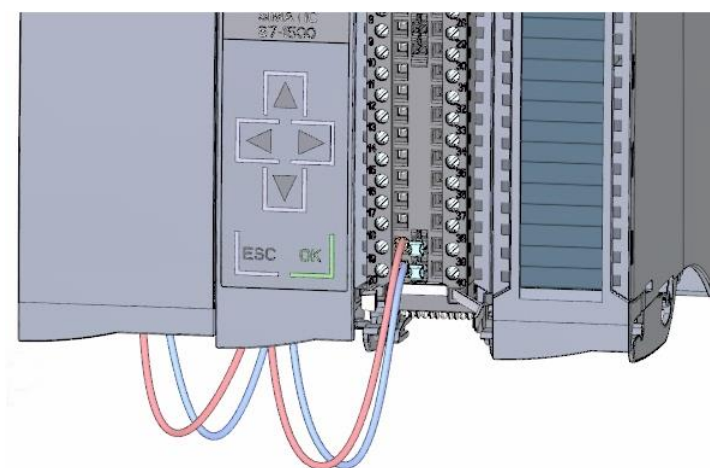
2. ケーブルタイに通します。



3. 電源電圧 24 V DC を端子 20 (M)と 19 (L+)に接続します。



4. 2つの底面の端子の間に電位回路ブリッジを挿入します。



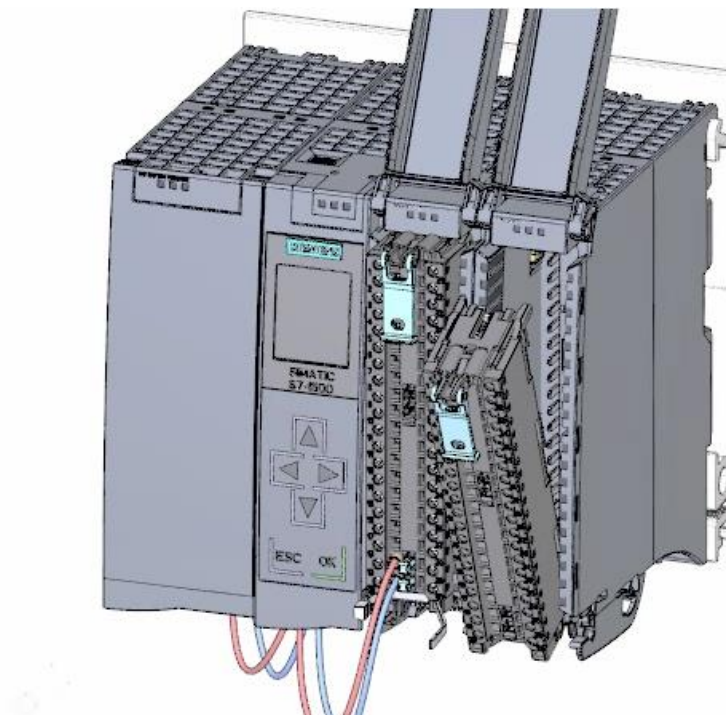
結果

デジタル入力モジュールがこれで配線されます。

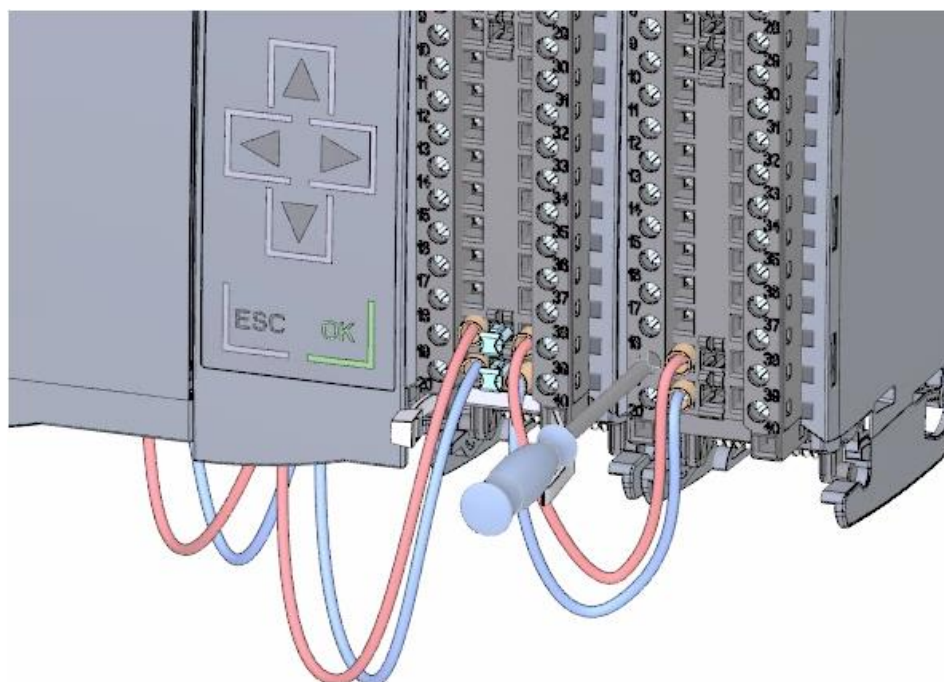
2.3.7 デジタル出力モジュールの配線

手順

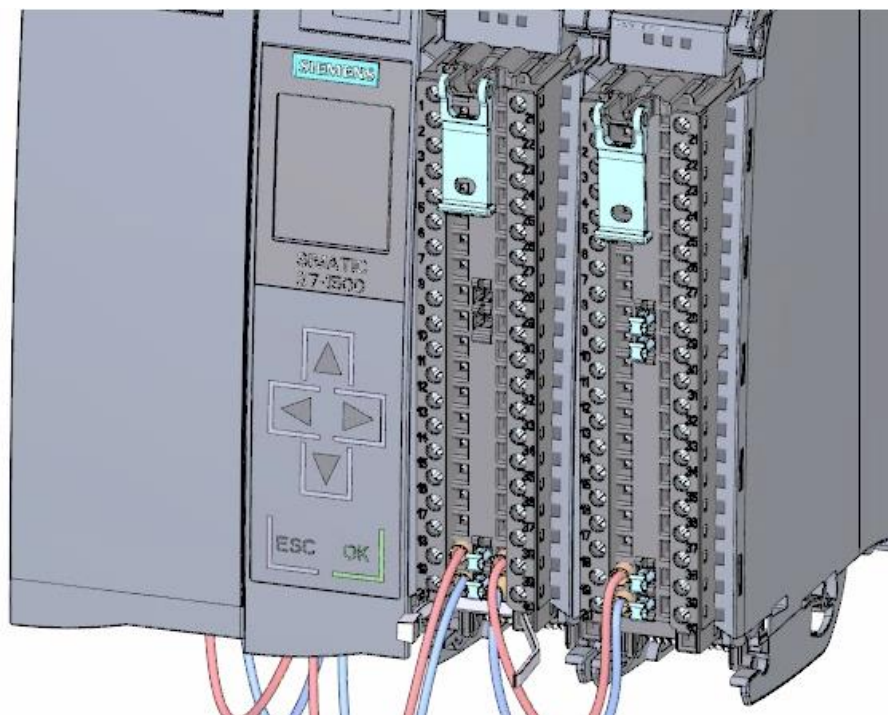
1. フロントコネクタを事前配線位置に挿入します。



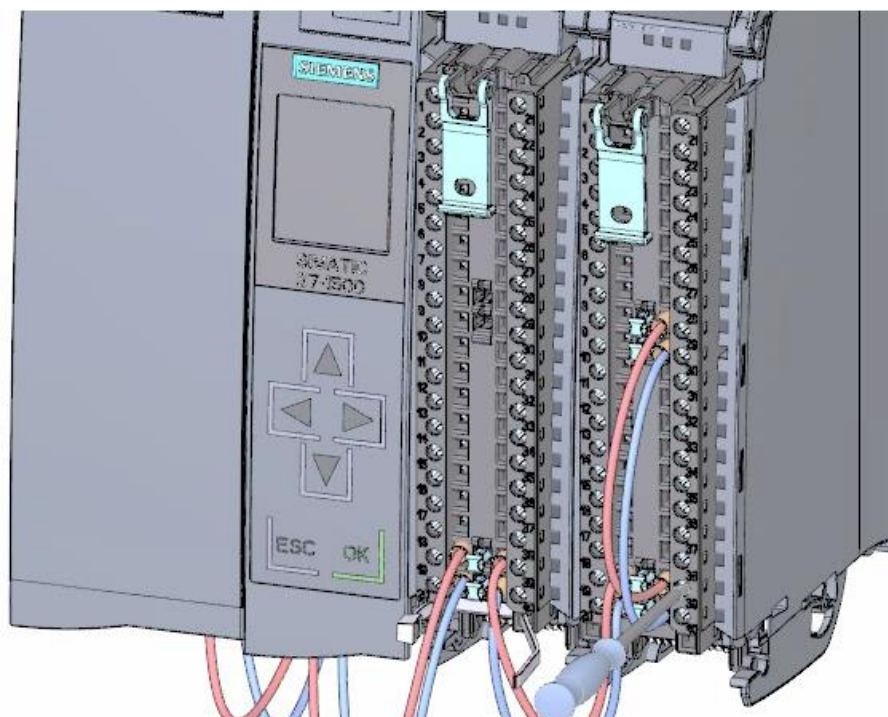
2. デジタル入力モジュールの端子 40 (M)と 39 (L+)を使用して、デジタル入力モジュールから端子 20 (M)と 19 (L+)に電源電圧 DC 24 V を送り込みます。



3. 4つの電位回路ブリッジを接続します。



4. 端子 30 と 40、同様に 29 と 39 を互いに接続します。



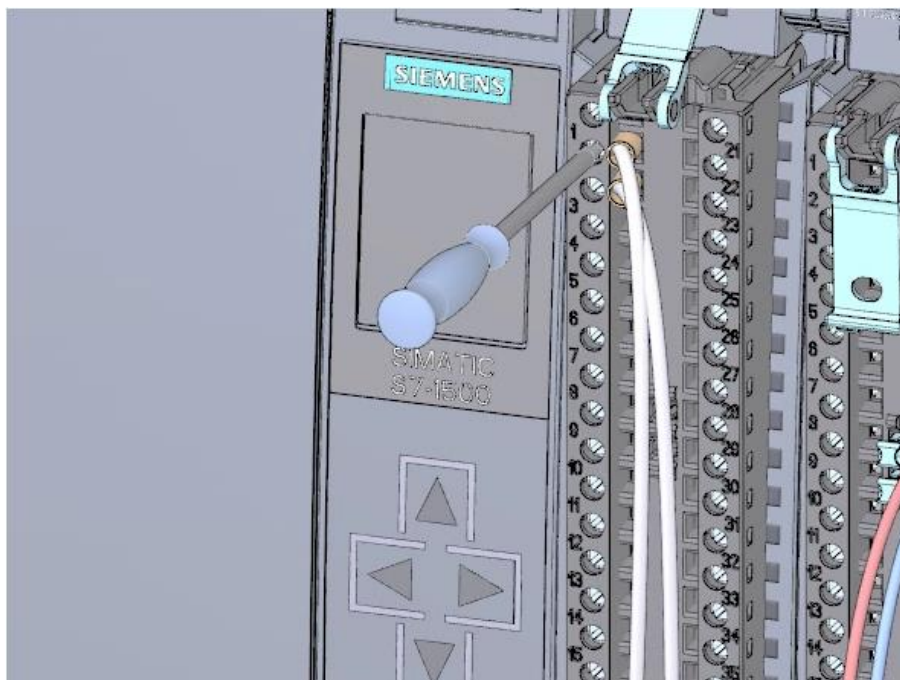
結果

デジタル出力モジュールがこれで配線されます。

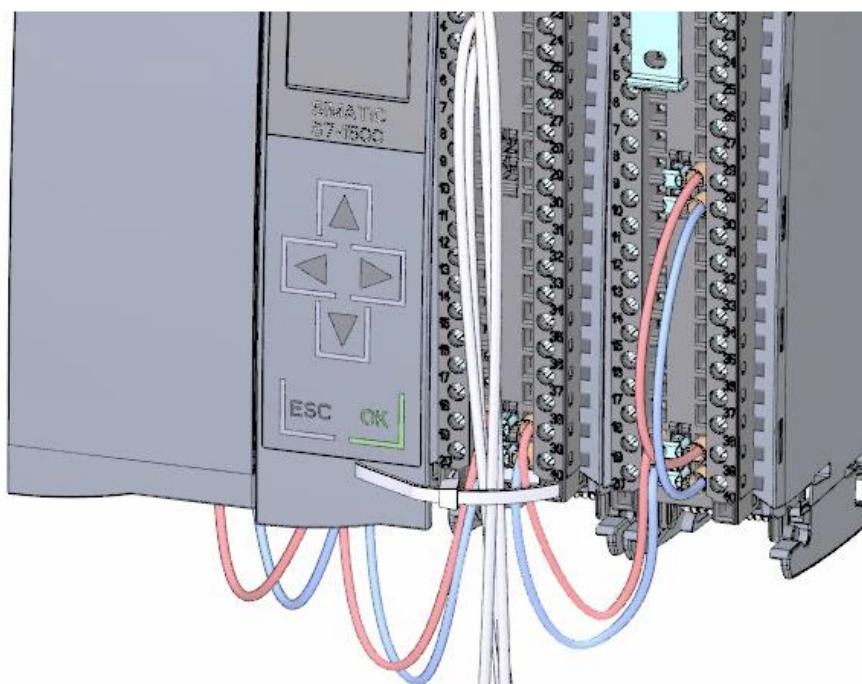
2.3.8 フロントコネクタの配線

手順

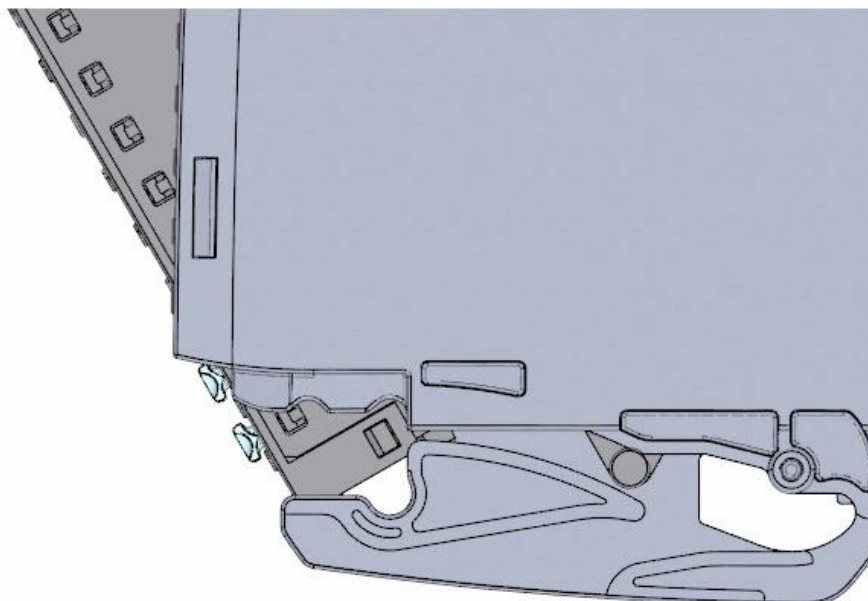
1. ターミナルのフロントカバーの内側の接続図に従って個々のワイヤを接続し、しっかりとネジ止めします。



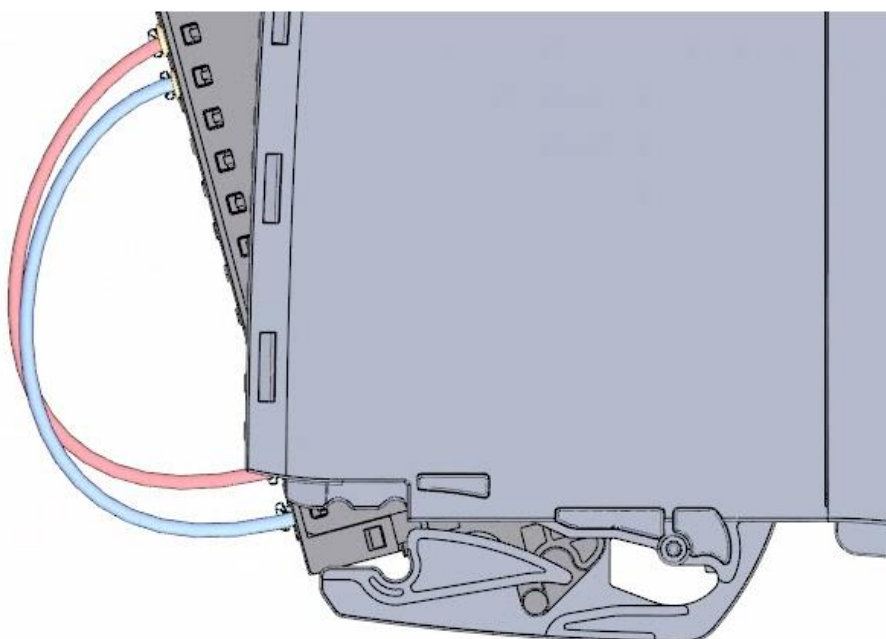
2. ストレインリリーフを使用する場合、ケーブルをケーブルハーネスの周りに巻き、しっかりと引っ張ります。



3. 事前の配線位置から最終的な位置まで、フロントコネクタを移動します。こうすることによって、フロントコネクタとモジュールの間の電氣的接続を作成します。



4. ヒント: モジュールの交換などの場合、事前配線済みのフロントコネクタは直接挿入できます。



結果

フロントコネクタがこれで配線されます。

2.4 電源オン

2.4.1 概要

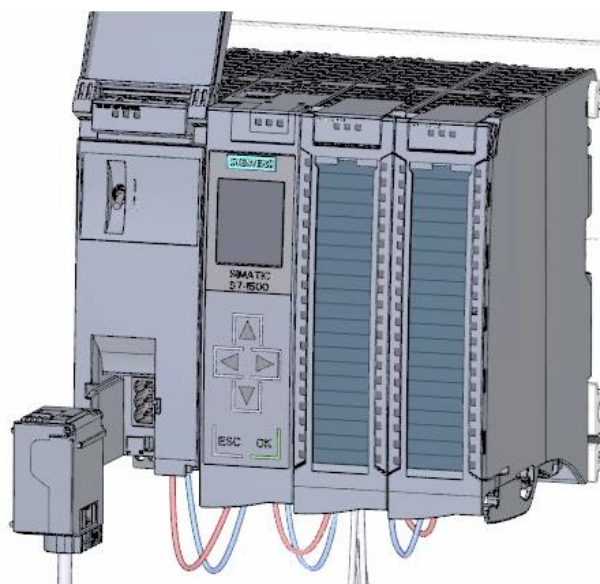
初めて CPU をオンにする

このセクションで、初めて CPU をオンにします。

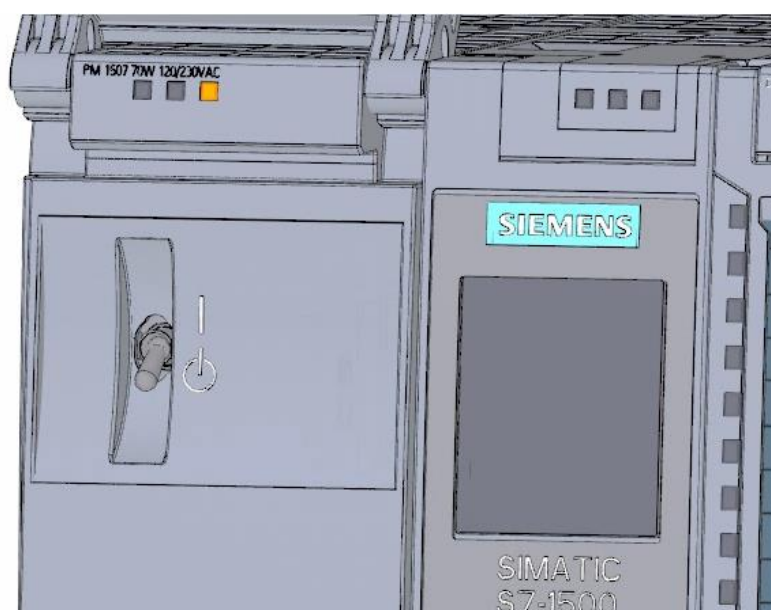
2.4.2 電源オン

手順

1. 電源モジュール(PM)の主電源接続プラグを挿入します。



2. 主電源接続プラグを電源に接続します。



3. 空の SIMATIC メモリカードを CPU に挿入します。



4. 電源モジュール(PM)のスイッチを位置 RUN にします。
CPU が起動します。



結果

CPU が起動し、STOP モードに入ります。

2.4.3 ディスプレイで IP アドレスを割り当て

この手順で CPU に IP アドレスとサブネットマスクを設定します。

手順

1. [Settings]に移動します。
2. [Addresses]を選択します。
3. インターフェース[X1 (IE/PN)]を選択します。
4. メニュー項目[IP Addresses]を選択します。
5. IP アドレス 192.168.0.10 を設定します。
6. モジュールの「右」矢印キーを押します。
7. サブネットマスク 255.255.255.0 を設定します。
8. モジュールの「下」矢印キーを押して、メニュー項目[Apply]を選択し、[OK]を押して選択を確定します。

結果

これで、インターフェース[X1 (IE/PN)]の IP アドレスとサブネットマスクを割り当てました。

ソフトウェアセクション

3.1 プロジェクトとハードウェアの作成

3.1.1 TIA ポータルの紹介

概要

Totally Integrated Automation ポータル(以降、TIA ポータルと呼ぶ)は、単一のクロスソフトウェアプラットフォームに構築される自動化タスクを実装するのに必要なすべてのファンクションを提供します。

TIA ポータルは、さまざまな SIMATIC システムを単一のフレームワークで利用できるようにした、統合エンジニアリングのための初の共同作業環境です。そのため、TIA ポータルにより、初めて信頼できる使い勝手の良いクロスシステムのコラボレーションが可能になりました。

すべての必要なソフトウェアパッケージ、ハードウェアコンフィグレーションとプログラミングからプロセスの可視化までが、包括的なエンジニアリングフレームワークに統合されています。



TIA ポータルで作業する利点

以下の機能は、TIA ポータルと連携して自動制御ソリューションを実現するための作業中に効率的なサポートを提供します。

- **統一された操作コンセプトによる統合エンジニアリング**

プロセスオートメーションとプロセスの可視化が「密接に関連」して行われます。

- **強力なエディタと汎用シンボルによる一貫した集中データ管理**

一度作成されたデータは、すべてのエディタで使用可能です。変更や訂正は、プロジェクト全体に自動的に適用されて、更新されます。

- **包括的なライブラリコンセプト**

既製の命令やプロジェクトの既存の部品を何度でも使用できます。

- **多数のプログラミング言語が使用可能**

自動化タスクを実装するのに、5種類のプログラミング言語を利用できます。

3.1.2 プロジェクトの作成

概要

次の手順で、新しいプロジェクトを作成します。

自動制御ソリューションの作成中に生成されたすべてのデータは、プロジェクトファイルに保存されます。データは、オブジェクト形式で保存されます。プロジェクト内で、オブジェクトはツリー構造で整理されます(プロジェクト階層)。

プロジェクトの階層は、構成データやプログラムと共にデバイスとステーションに基づいています。

必要条件

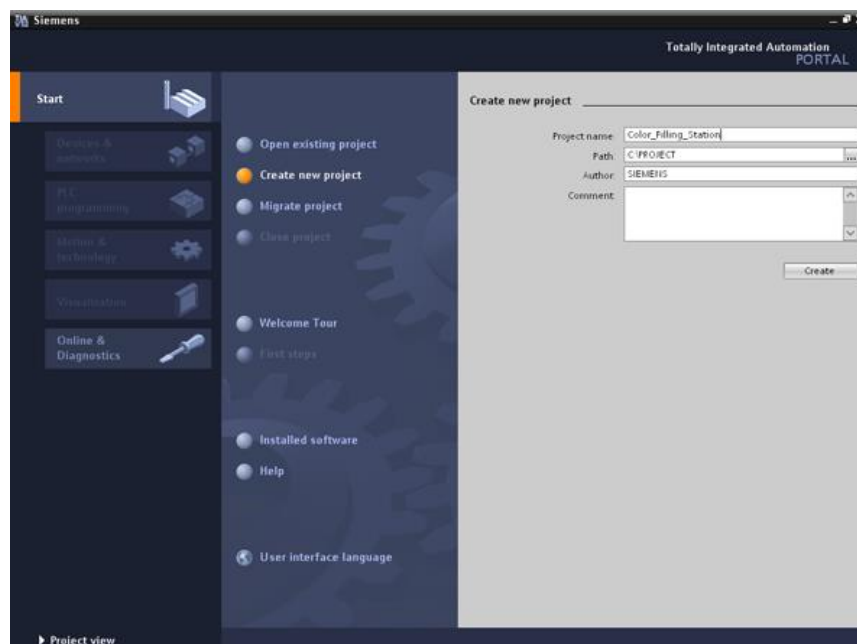
プロジェクトの作成には、以下のハードウェア機器とソフトウェアの装備が必要です。

- ハードウェア:
 - CPU 1511-1 PN、入門書のハードウェアセクションに従って取り付けおよび配線済み
 - プログラミングツール/PC へのイーサネット接続
- ソフトウェア:
 - 以下のソフトウェアパッケージが、プログラミングツール/PC にインストールされ、実行可能でなければなりません。
 - SIMATIC STEP 7 Professional V13
 - SIMATIC WinCC Advanced V13 または SIMATIC WinCC Professional V13

新しいプロジェクトの作成

新しいプロジェクトを作成するには、次の手順を実行します。

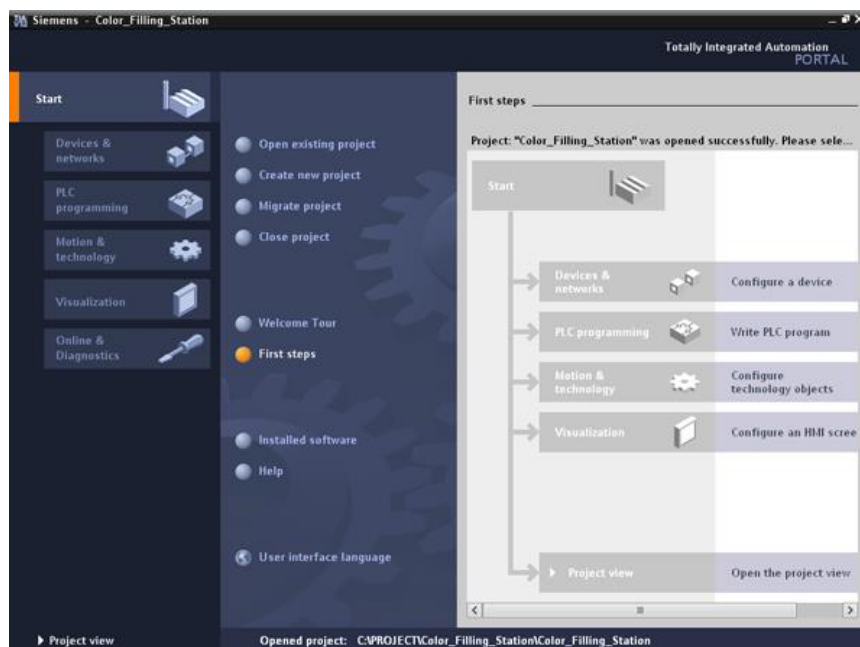
1. [Create new project]をクリックします。
2. プロジェクト名を入力します。



3. [Create]をクリックして、新しいプロジェクトを作成します。

結果

プロジェクトが作成されました。ハードウェアコンフィグレーション、CPU プログラミング、および HMI でのビジュアライゼーションのようなすべてのデータがプロジェクトに保存されます。



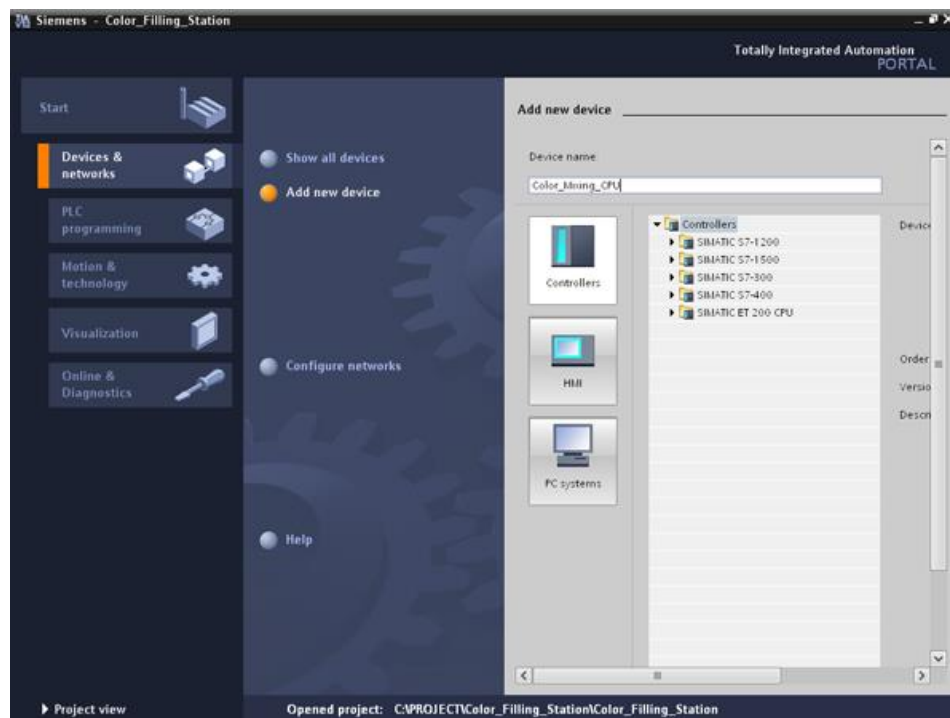
3.1.3 S7-1500 CPU の作成

概要

以下の手順で、未指定の CPU を作成します。未指定の CPU は、後で定義されるハードウェアカタログからの特定の CPU 用のプレースホルダになります。

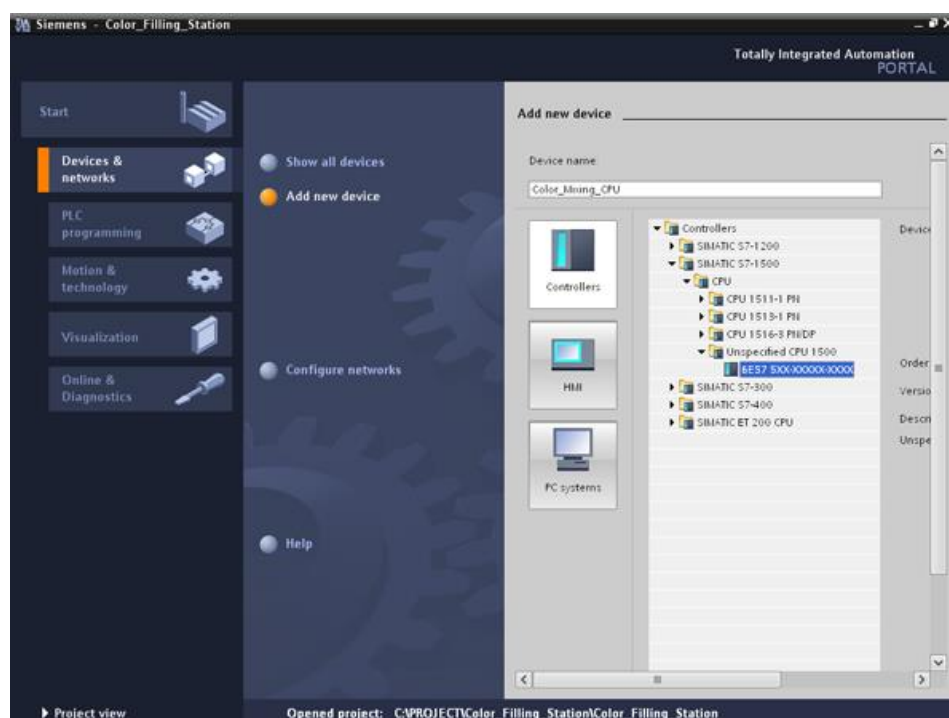
手順

1. [Devices & Networks]ポータルを開きます。
2. [Add new device]をクリックします。
3. CPU の名前として、[Color_Mixing_CPU]を入力します。



4. [SIMATIC S7-1500]フォルダを開きます。

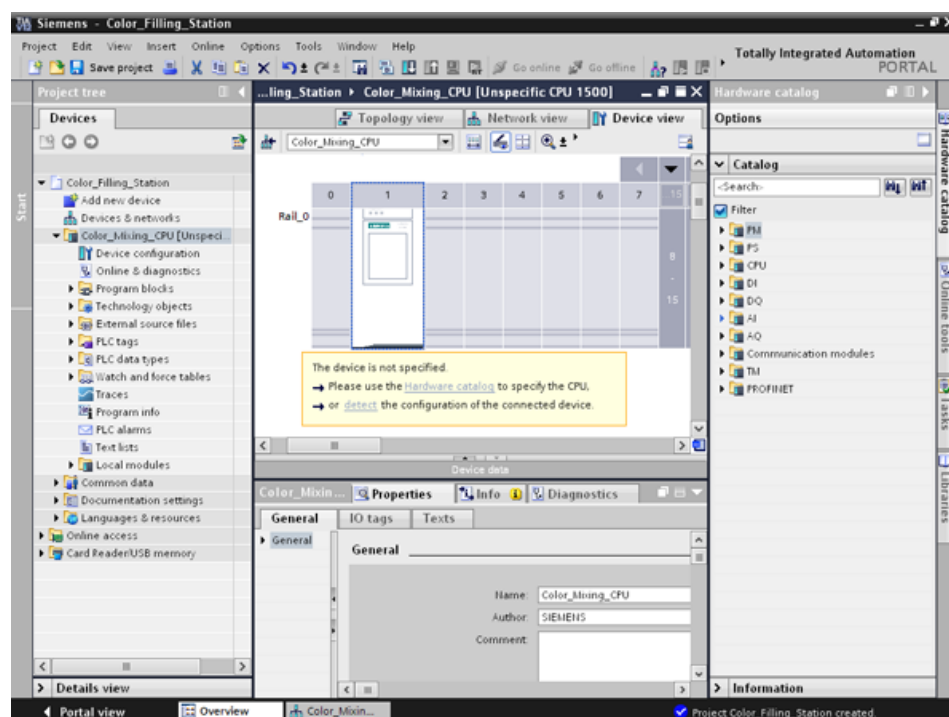
5. Unspecified CPU を選択します。



6. ダブルクリックして CPU を作成します。

結果

未指定の CPU がプロジェクトファイルに作成されます。この CPU のためのユーザープログラムのコンテンツをこの時点で既に作成できます。



3.1.4 ハードウェア検出の実行

概要

以下のセクションで、CPU タイプを読み取るためにハードウェア検出ファンクションを使用します。

ハードウェア検出中に LED 点滅テストを実施します。LED 点滅テストは、検出されたデバイスの LED を有効にします。また、このファンクションを利用して、複数のデバイスから構成されるハードウェアコンフィグレーションで正しいデバイスが選択されているかを検証できます。

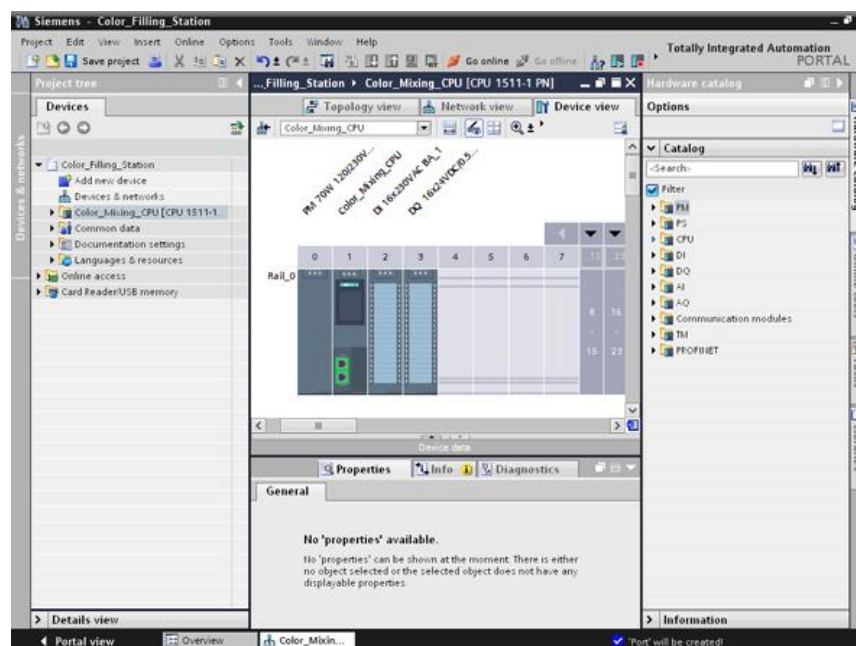
手順

1. プロジェクトツリーで未指定の CPU を選択します。
2. [Online]メニューから[ハードウェア検出]ファンクションを選択します。
オプション 2: デバイスビューの黄色の枠のついたアラームをクリックします。
3. PG/PC インターフェースのタイプとして[PN/IE]エントリを選択します。
4. PG/PC インターフェースを選択します。
5. [Show all compatible devices]オプションをクリックします。
6. サブネットの互換デバイスから CPU を選択します。
7. [Flash LED]チェックボックスを選択して、点滅テストを実行します。
8. [Detect]をクリックして未指定の CPU を必要な CPU タイプと置き換えます。

結果

CPU タイプが読み出されます。正しいデバイス名が、プロジェクトツリーの CPU 名の括弧の中に追加されます。

使用する CPU とモジュールが、ハードウェアコンフィグレーションに表示されます。



3.1.5 ET 200 インターフェースモジュールの作成

概要

以下のセクションで、ハードウェアコンフィグレーションに 2 つのリモート I/O システムを作成します。

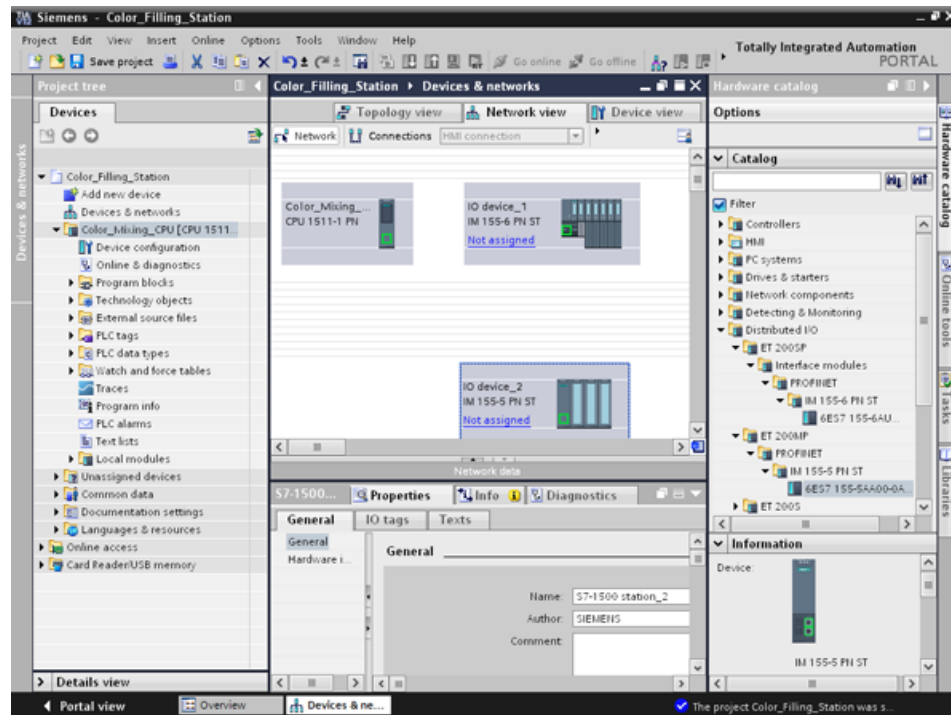
- ET 200SP リモート I/O システムは、基本的に以下のコンポーネントから構成されます。
 - CPU と通信するためのインターフェースモジュール
 - どの組み合わせでも挿入可能な最大 32 台のモジュール
 - コンフィグレーションを完成するサーバーモジュール
- ET 200MP リモート I/O システムは、以下のコンポーネントから構成されます。
 - CPU と通信するためのインターフェースモジュール
 - 最大 30 台のモジュール、各モジュールは最大 32 チャンネルを提供

手順

1. [Hardware catalog]を開きます。
2. [Network view]に変更します。
3. [Distributed I/O]を開き、[ET 200SP]フォルダを開きます。
4. [IM 155-6 PN ST]フォルダを開きます。
5. [6ES7 155-6AU00-0BN0]インターフェースモジュールをネットワークビューにドラッグ & ドロップします。
6. [ET 200MP]フォルダを開きます。
7. [IM 155-5 PN ST]フォルダを開きます。
8. [6ES7 155-5AA00-0AB0]インターフェースモジュールをネットワークビューにドラッグ & ドロップします。

結果

I/O システムがハードウェアコンフィグレーションに作成されましたが、まだ CPU 1511-1 PN に割り当てられていません。I/O システムは両方とも、プロジェクトビューの[Unassigned devices]の下に表示されます。



追加情報

SIMATIC ET 200 製品ファミリーは、お客様固有のアプリケーションに適合する各種の拡張可能な I/O システムを提供します。

SIMATIC ET 200 リモート I/O の詳細については、インターネットを参照してください (<https://new.siemens.com/jp/ja/products/automation/systems/industrial/io-systems.html>)。

3.1.6 ET 200 インターフェースモジュールのネットワーク接続

概要

以下のセクションで、PROFINET I/O システムを作成します。

PROFINET I/O システムは、PROFINET IO コントローラとその関連 PROFINET IO デバイスで構成されます。

- 既に作成された CPU 1511-1 PN が、PROFINET IO コントローラとして使用されます。
- 2 つのリモート I/O システムは、PROFINET IO デバイスとして使用されます。

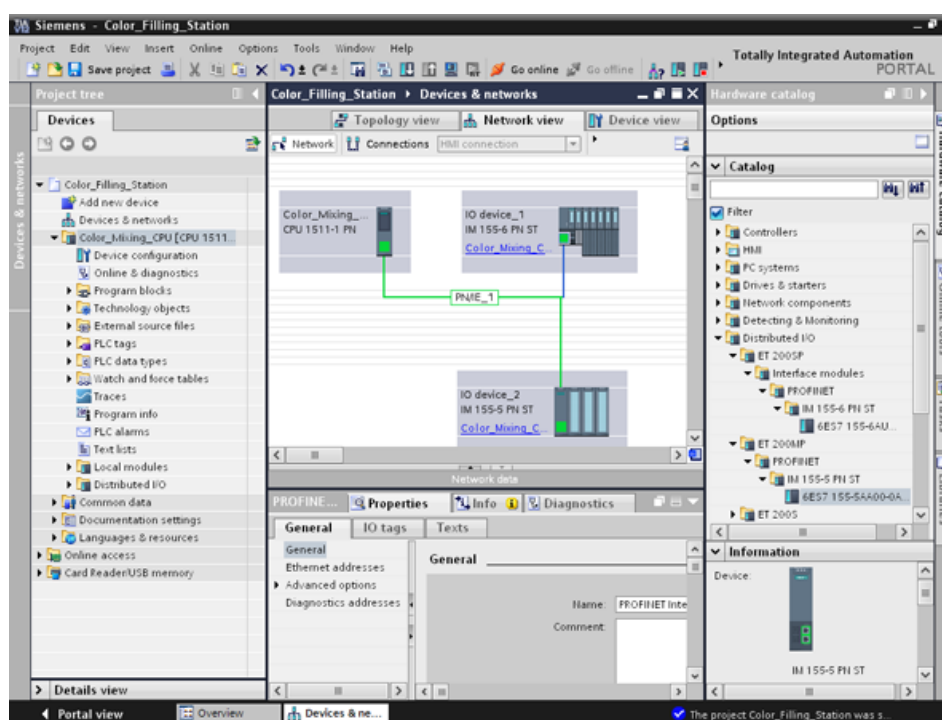
手順

1. IM 155-5 PN ST インターフェースモジュールのインターフェースから CPU インターフェースへ、接続をドラッグ&ドロップします。
2. IM 155-6 PN ST インターフェースモジュールと CPU の間の 2 番目の接続を作成します。

結果

インターフェースモジュールが、IO デバイスとして CPU に割り当てられます。リモート IO システムが両方とも、CPU の下の[Distributed I/O]フォルダのプロジェクトツリーに表示されます。

PROFINET I/O システムがネットワーク接続プロセスで自動的に作成され、そのプロパティがネットワークビューに表示されます。



3.1.7 ET 200SP 用の入出力モジュールとサーバーモジュールの作成

概要

以下のセクションで、ET 200SP 用に入出力モジュールを作成します。

注記

入出力モジュールを動作するにはサーバーモジュールが必要です。これらのモジュールは、サーバーモジュールがない場合、動作しません。

電源グループ当たりの最大コンフィグレーション

電源グループごとに使用できる I/O モジュール数は、以下の要因に依存します。

1. この電源グループで動作するすべての I/O モジュールに必要な合計電力
 2. この電源グループに外部的に接続されているすべての負荷に必要な合計電力
- 1 と 2 から計算された合計電力の総計は、使用される BaseUnit と負荷電源電圧の通電容量を超えることはできません。

モジュールの[Potential group]パラメータを以下のように設定します。

パラメータ	値の範囲	使用
電源グループ	左のモジュールの電源グループを使用 (デフォルト設定)	左のすべてのモジュールの合計消費電力 + モジュールの消費電力が、BaseUnit の通電容量よりも小さい場合
	新しい電源グループの有効化	左のすべてのモジュールの合計消費電力 + モジュールの消費電力が、BaseUnit の通電容量よりも大きい場合

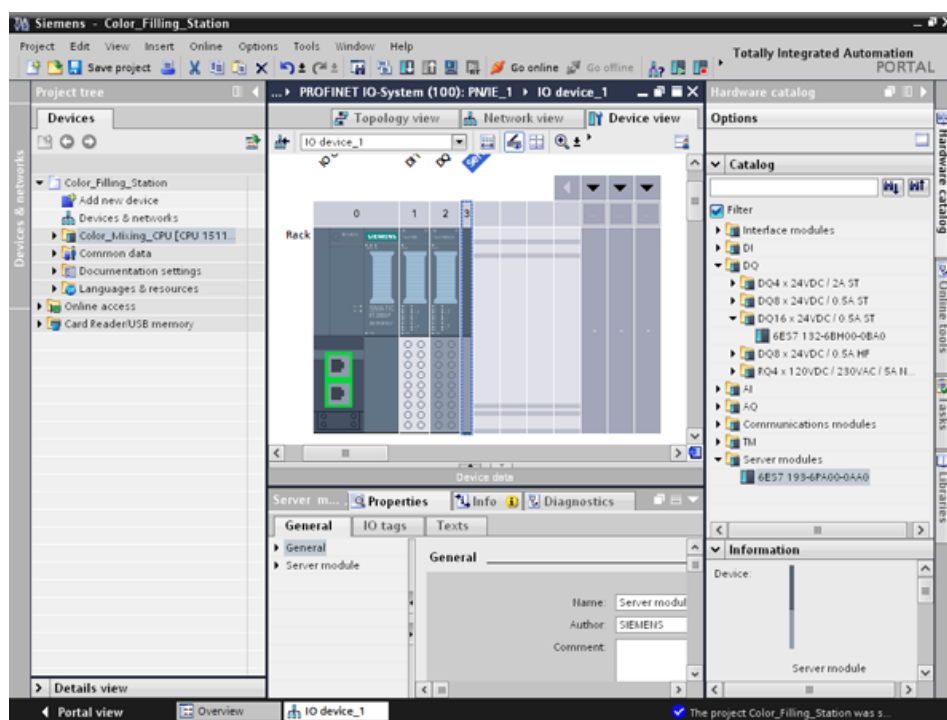
『SIMATIC ET 200SP DI 8x24VDC HF デジタル入力モジュール』のようなモジュールマニュアルに電源グループについての詳しい情報が記載されています (<http://support.automation.siemens.com/DE/view/en/66912542>)。

手順

1. ET 200SP のデバイスビューを開きます。
2. ハードウェアカタログの[DI]および[DI16 x DC24V ST]フォルダを開きます。
3. レールのスロット 1 に、入力モジュール[6ES7 131-6BH00-0BA0]をドラッグ&ドロップします。
4. [DQ]と[DQ16 x DC24V / 0.5A ST]フォルダを開きます。
5. レールのスロット 2 に、出力モジュール[6ES7 132-6BH00-0BA0]をドラッグ&ドロップします。
6. [Server modules]フォルダを開きます。
7. レールのスロット 3 に、サーバーモジュール[6ES7 193-6PA00-0AA0]をドラッグ&ドロップします。

結果

入出力モジュールとサーバーモジュールを作成しました。



3.1.8 ET 200MP 用の入出力モジュールの作成

概要

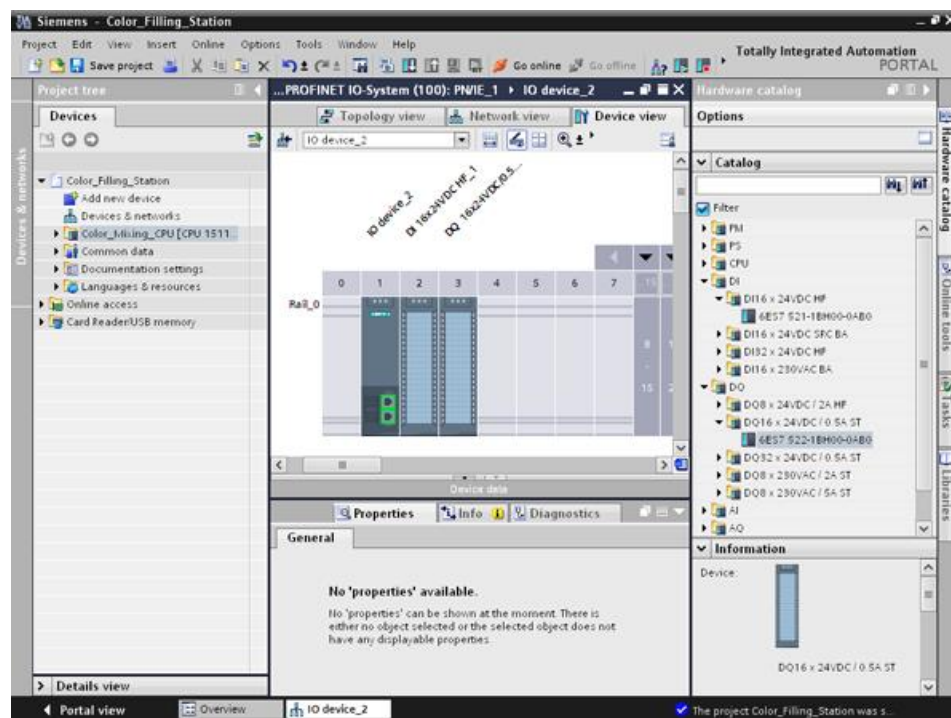
以下のセクションで、ET 200MP 用の入出力モジュールを作成します。

手順

1. ET 200MP のデバイスビューを開きます。
2. ハードウェアカタログの[DI]および[DI16 x DC24V HF]フォルダを開きます。
3. レールのスロット 2 に、入力モジュール[6ES7 521-1BH00-0AB0]をドラッグ&ドロップします。
4. [DQ]と[DQ16 x DC24V / 0.5A ST]フォルダを開きます。
5. レールのスロット 3 に、出力モジュール[6ES7 522-1BH00-0AB0]をドラッグ&ドロップします。

結果

入出力モジュールを作成しました。



3.1.9 ET 200 に名前を割り当てる

概要

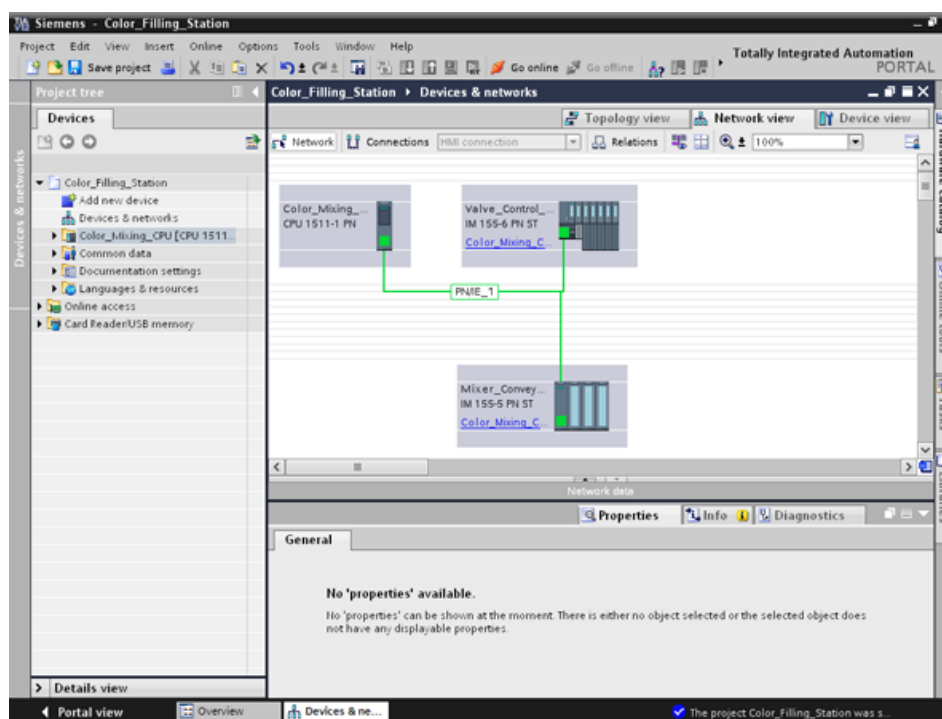
以下のセクションで、プロジェクト固有の名前をリモート I/O に割り当てます。

手順

1. ET 200SP を選択します。
2. インспекタウィンドウの[Properties] → [General]で、[Name]フィールドに名前 [Valve_Control_Unit]を入力します。
3. ET 200MP を選択し、新しい名前[Mixer_Conveyor_Control_Unit]を入力します。

結果

プロジェクト固有の名前を割り当てました。



3.2 プログラムの作成

3.2.1 ブロックライブラリのロード

概要

以下のセクションで、グローバルライブラリ[ProgLib_ColorFillingStation]をロードします。このライブラリには、プロジェクト例に必要なブロックとタグテーブルが含まれています。このライブラリは、「Getting Started S7-1500 / TIA V13 (http://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/project/color_filling_station.zip)」にある ZIP ファイルで利用できます。プロジェクトにインポートする前に、このライブラリを解凍する必要があります。

グローバルライブラリ

グローバルライブラリは、他のプロジェクトで再利用したいエレメントを保管するのに使用します。グローバルライブラリは、明示的に作成する必要があります。

以下のライブラリは標準パッケージで提供されます。

- [Buttons and Switches]

多数のスイッチとボタンの選択肢を提供します。フォルダは、スイッチとボタンをカテゴリ別に整理します。たとえば、[DiagnosticsButtons]フォルダで[System diagnostics indicator]オブジェクトを見つけることができます。プラントでシステム診断のために[System diagnostics indicator]オブジェクトを使用できます。

- [Monitoring and Control objects]

これは、適切なコントロールライト、ボタンおよびスイッチと同様に、複雑なオペレータコントロールと表示オブジェクトをいくつかのデザインで提供します。

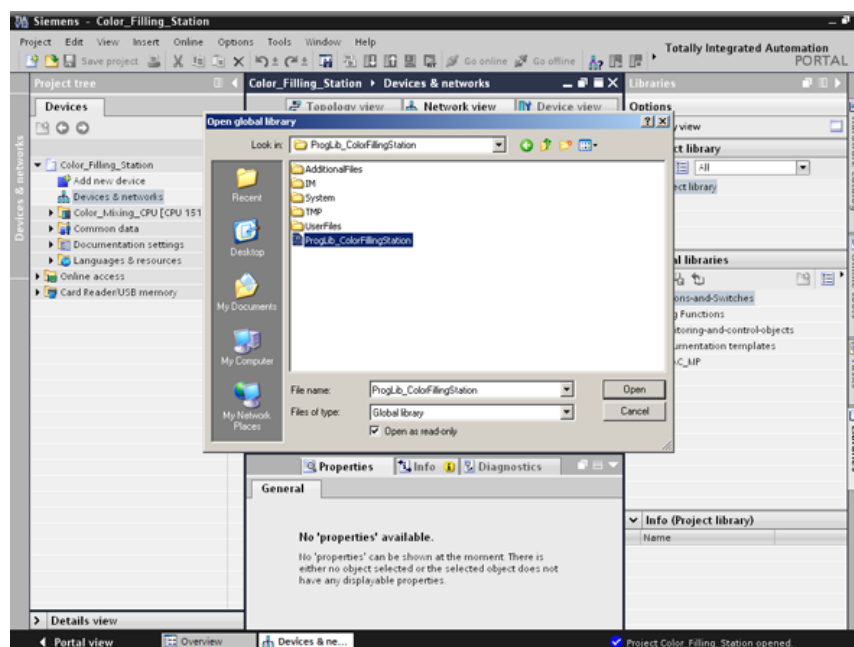
注記

書き込み保護されたライブラリ

[Open read-only]オプションは、デフォルトで[Open global library]ダイアログで有効です。チェックボックスをクリックして、書き込み保護のないライブラリを開きます。

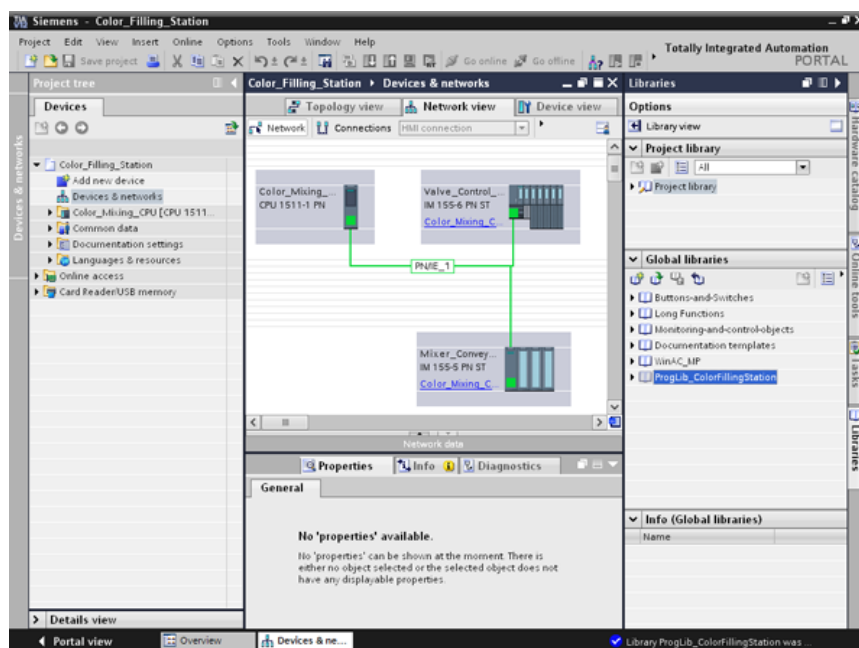
手順

1. [Libraries]タブをクリックします。
2. [Open global library]をクリックします。
3. 解凍されたライブラリフォルダを含むディレクトリから[ProgLib_ColorFillingStation]ファイルを選択し、[Open]をクリックします。



結果

[ProgLib_ColorFillingStation]グローバルライブラリが開きます。

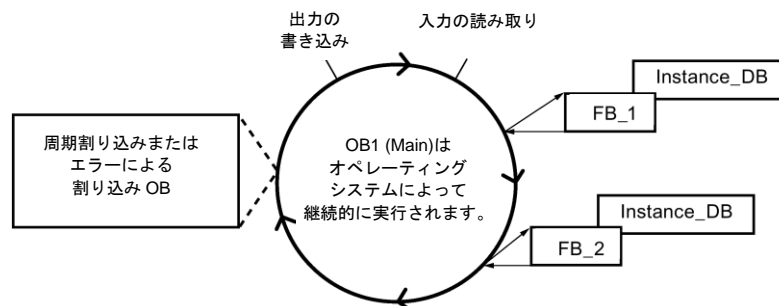


3.2.2 プログラムブロック Main [OB1]の削除

概要

以下のセクションで、プロジェクトフォルダから自動的に生成された[Main [OB1]]プログラムブロックを削除します。[Main [OB1]]プログラムブロックは、プロジェクト例のプログラムブロックに含まれています。

オーガニゼーションブロック(OB)は、CPU オペレーティングシステムとユーザープログラムの間のインターフェースを形成します。これらのブロックは、オペレーティングシステムによって呼び出されます。少なくとも 1 サイクル OB が、オートメーションプロジェクトで使用できなければなりません。

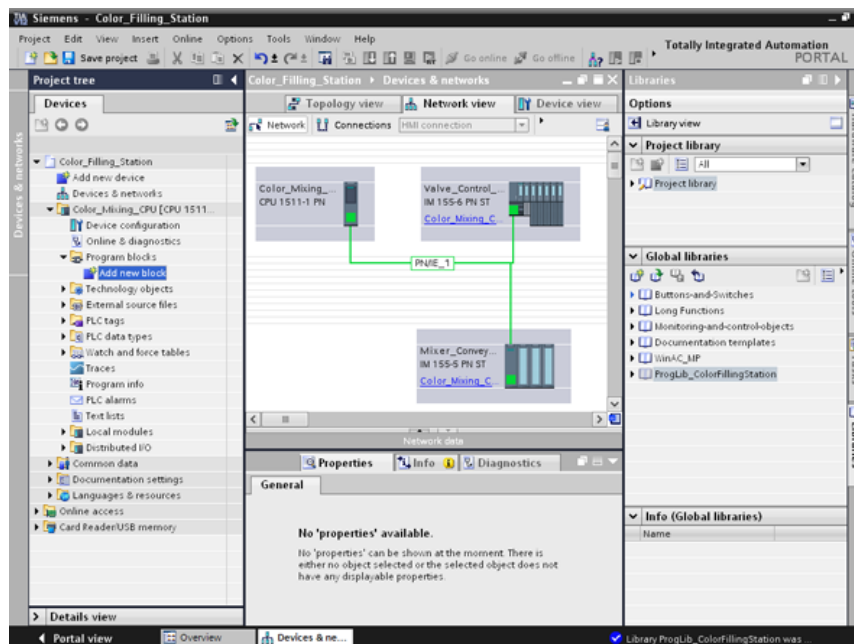


手順

1. プロジェクトツリーで[Program blocks]フォルダを開き、次に[Main [OB1]]プログラムブロックをクリックします。
2. 右クリックして、ショートカットメニューを開き、次に[Delete]をクリックします。
3. [Yes]をクリックして、ブロックの削除を確定します。

結果

自動的に生成された[Main [OB1]]プログラムブロックが削除されます。



3.2.3 プログラムブロックのコピー

概要

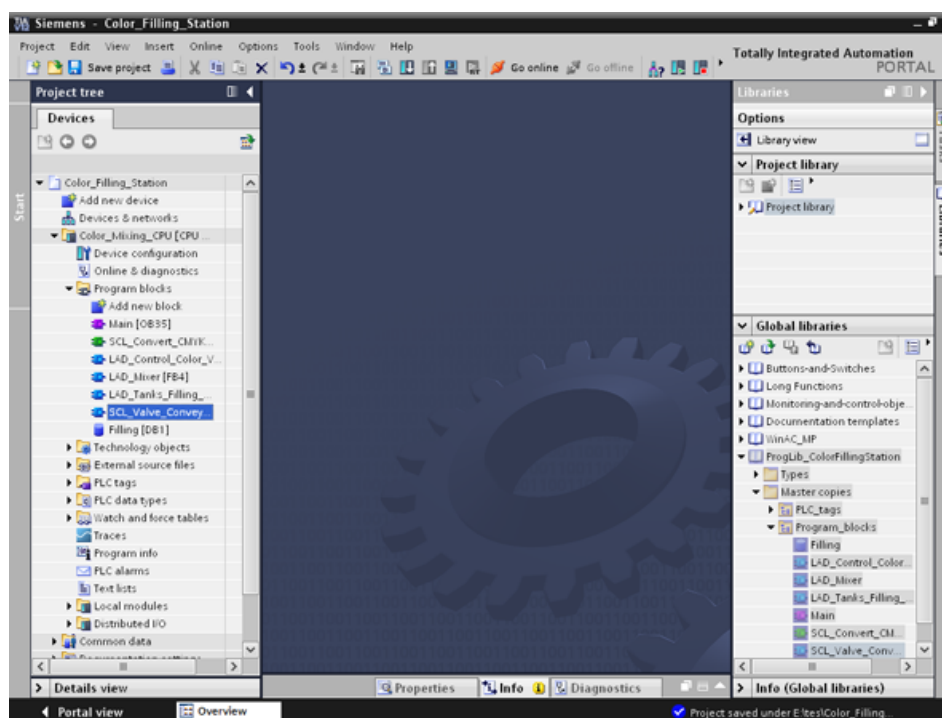
以下のセクションで、[ProgLib_ColorFillingStation]グローバルライブラリからプロジェクトにプログラムブロックを挿入します。

手順

1. [ProgLib_ColorFillingStation]グローバルライブラリをクリックします。
2. [Master copies]フォルダをクリックして、次に[Programm_blocks]をクリックします。
3. グローバルライブラリから[Program blocks]フォルダへ、インポートされるプログラムブロックをドラッグ&ドロップします。
4. 他のブロックの場合も、手順 2 および 3 で記載したように進めます。

結果

プログラムブロックが、同じ名前のプロジェクトフォルダに挿入されます。



3.2.4 周期割り込み OB

3.2.4.1 周期割り込み OB – サイクルタイムと位相

サイクルタイムと位相オフセットは変更可能

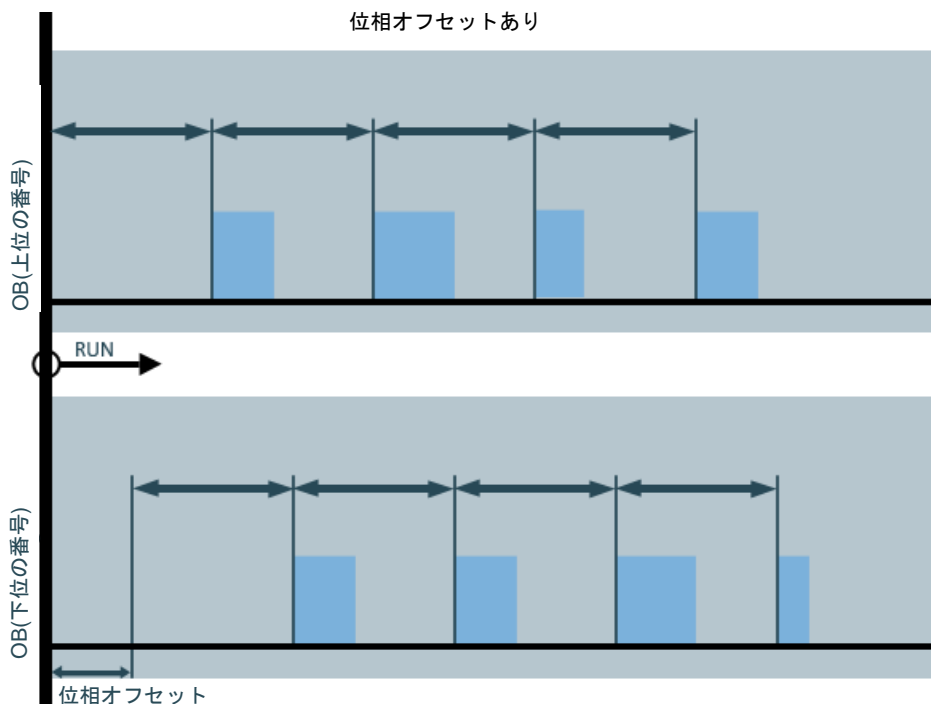
Main [OB35]は、プロジェクトに挿入されるプログラムブロックの下にあります。
Main [OB35]は、周期割り込みオーガニゼーションブロック(周期割り込み OB)です。周期割り込み OB は、サイクリックなプログラムの実行とは無関係に、定期的な時間間隔でプログラムを開始するのに使用します。周期割り込み OB の開始時間は、サイクルタイムと位相オフセットを使用して指定します。

サイクルタイム

サイクルタイムは、OB を呼び出す間隔を確定します。周期割り込み OB のサイクルタイムは、デフォルトで 100000 μ s です。

位相オフセット

位相オフセットは、周期割り込みプログラムの処理間隔の精度を上げるのに使用します。OB が他の OB と同じ、または共通の複数クロックパルスを持っている場合、両者は位相オフセットで正確な間隔で動作できます。



3.2.4.2 サイクルタイムの変更

概要

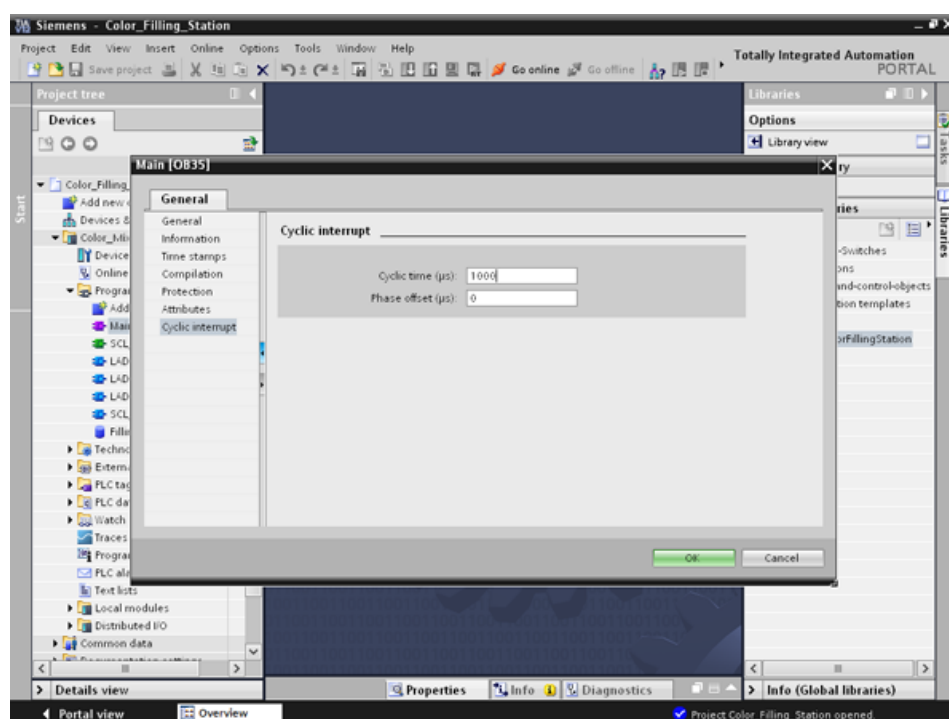
以下のセクションで、[Main]プログラムブロックのサイクルタイムを変更します。

必要条件

- プログラムブロック[Main] [OB35]がライブラリに含まれている
- FB/FC 呼び出しが存在

手順

1. [Main]プログラムブロックのプロパティを開きます。
2. [General]で[Cyclic interrupt]オプションを選択します。
3. [Cycle time]に新しい値を入力し、[OK]をクリックします。



結果

サイクルタイムが変更されます。

3.2.5 タグテーブルのコピー

概要

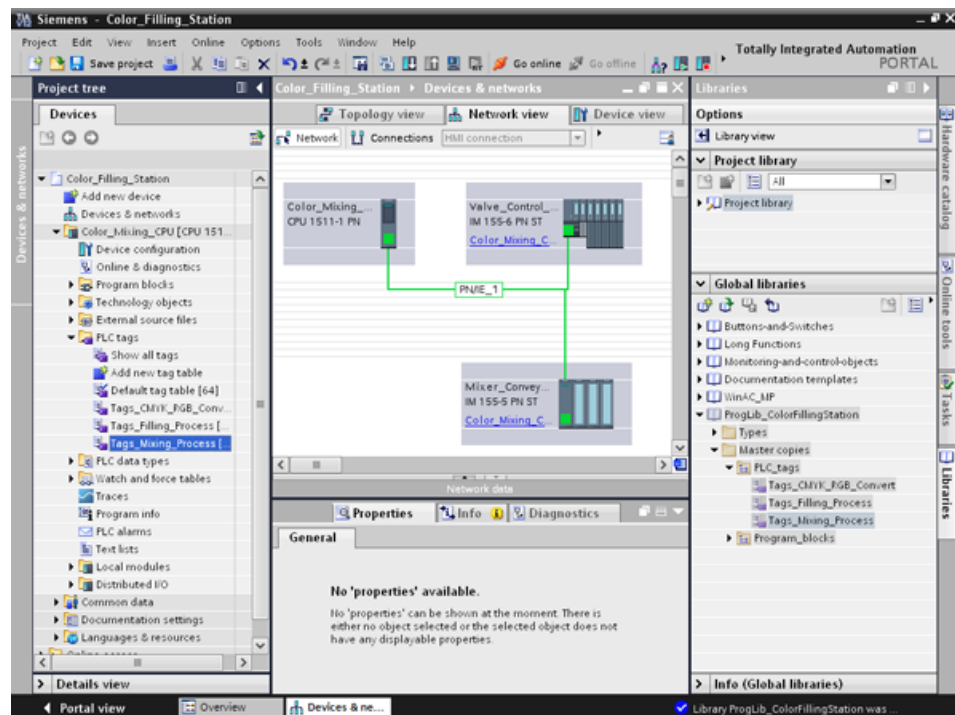
以下のセクションで、[ProgLib_ColorFillingStation]グローバルライブラリからプロジェクトに、タグテーブルを挿入します。

手順

1. プロジェクトナビゲーションで[PLC tags]フォルダを開きます。
2. [PLC_tags]フォルダを開きます。
3. グローバルライブラリから[PLC tags]フォルダへ、インポートされるタグテーブルをドラッグ&ドロップします。
4. 他のタグテーブルの場合も、手順3で記載したように進めます。

結果

タグテーブルが、同じ名前のプロジェクトフォルダに挿入されます。



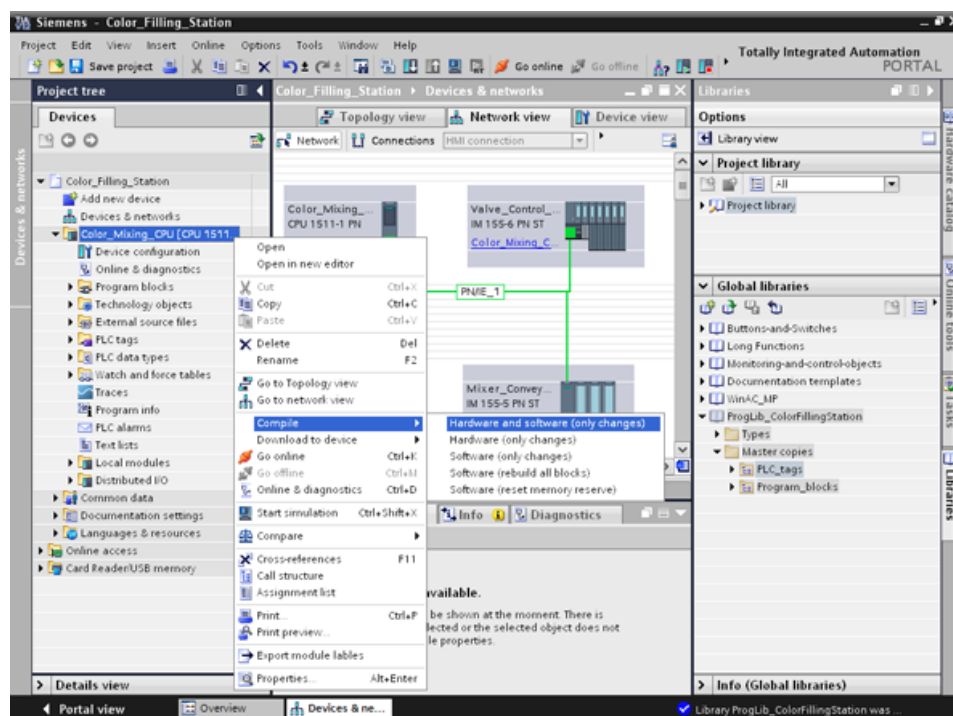
3.2.6 プロジェクトのコンパイル

概要

次のセクションで、[Color_Filling_Station]プロジェクトをコンパイルします。

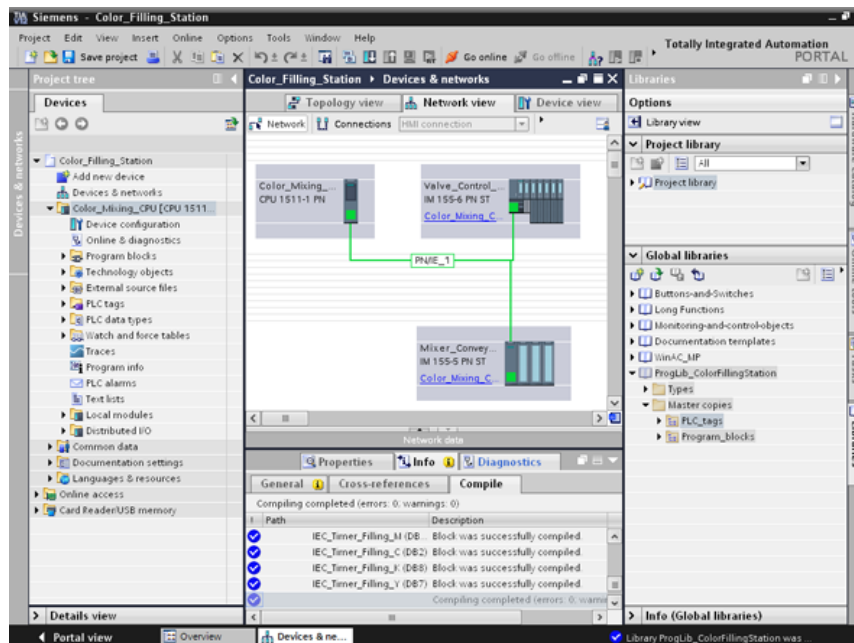
手順

1. プロジェクトツリーで[Color_Mixing_CPU]という CPU を選択します。
2. 右クリックして、ショートカットメニューを開き、次に[Compile] → [Hardware and software (only changes)]を選択します。



結果

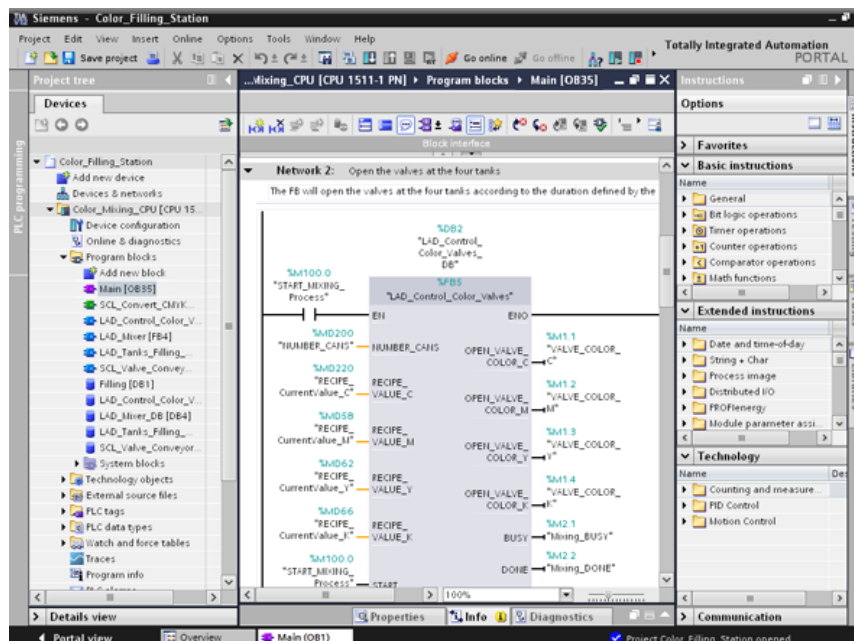
プロジェクトがコンパイルされ、ダウンロードの準備が完了しました。



注記

[Main]プログラムブロックは更新されます。

コンパイル後、[Main]プログラムブロックを開きます。すべてのインスタンスデータブロックが作成され、データブロックが更新されます。



3.2.7 プロジェクトを CPU にダウンロード

概要

次のセクションで、[Color_Filling_Station]プロジェクトを CPU にダウンロードします。

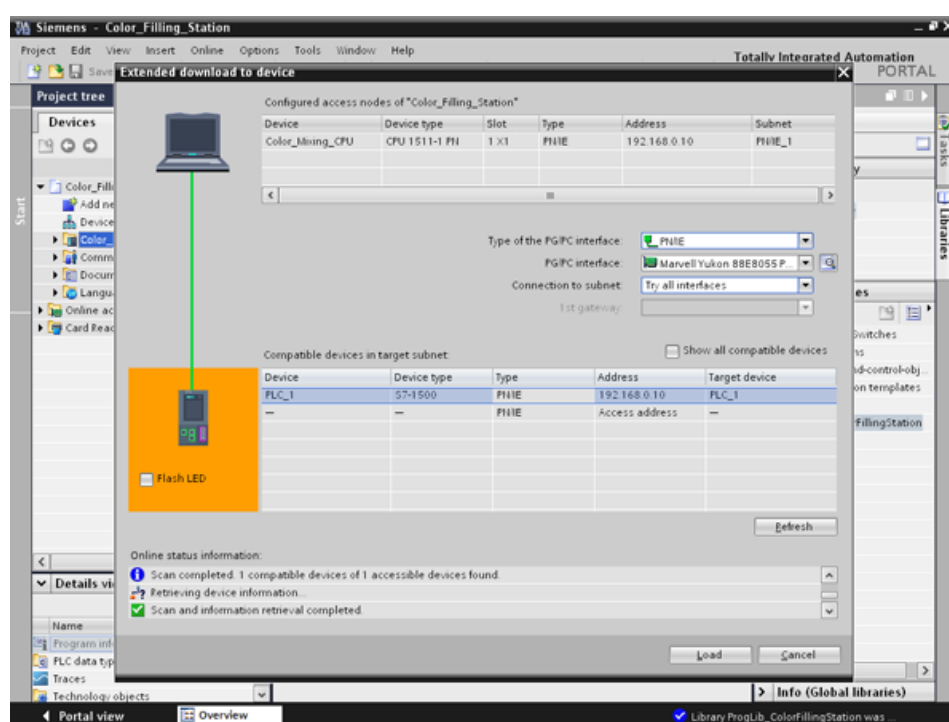
注記

すべての互換デバイスの表示

[Extended download to device]ダイアログで設定した後で必要な CPU が表示されない場合、オプション[Show all compatible devices]をクリックします。

手順

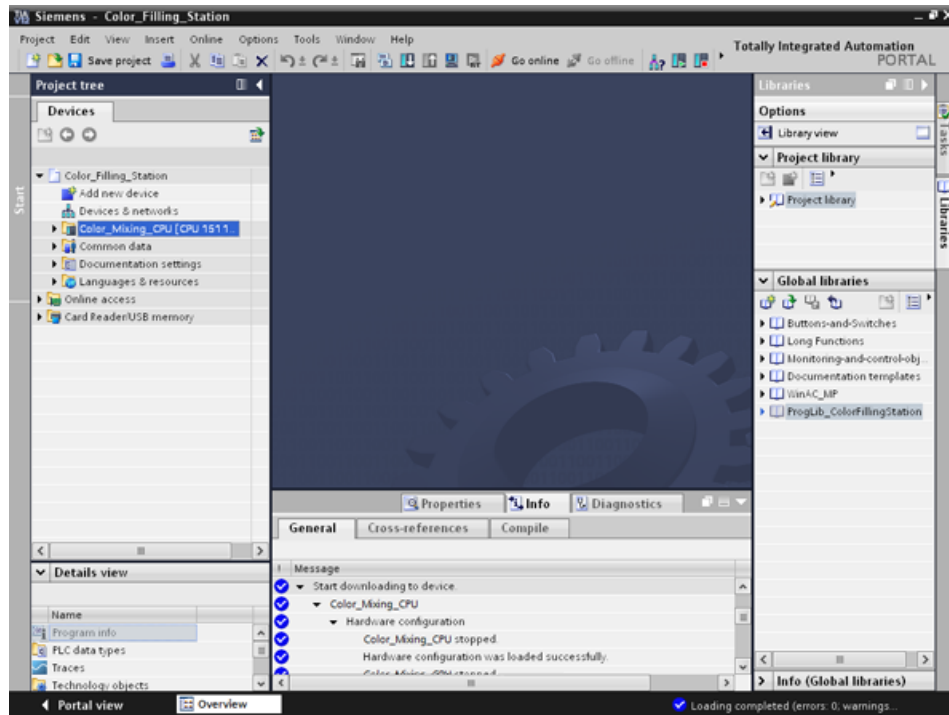
1. CPU ショートカットメニューを開き、[Download to device] → [Hardware and software (only changes)]を選択します。
2. ドロップダウンリストから、PG/PC インターフェースタイプ、インターフェース、およびサブネットとの接続を選択します。
3. サブネットの互換デバイスから CPU を選択し、[Load]を選択します。



4. 2つの[Assign IP address]ダイアログを[Yes]と[OK]で確定します。
5. [Load preview]ダイアログで、ドロップダウンリストの[No action]に設定されているすべてのエントリに対して別のエントリを選択し、開いているオプションを確定します。
6. [Load]をクリックします。
7. [Start all]オプションを確定し、[Finish]をクリックします。

結果

プロジェクトがCPUにダウンロードされます。



3.2.8 最適化ブロックアクセス

3.2.8.1 概要

操作原理

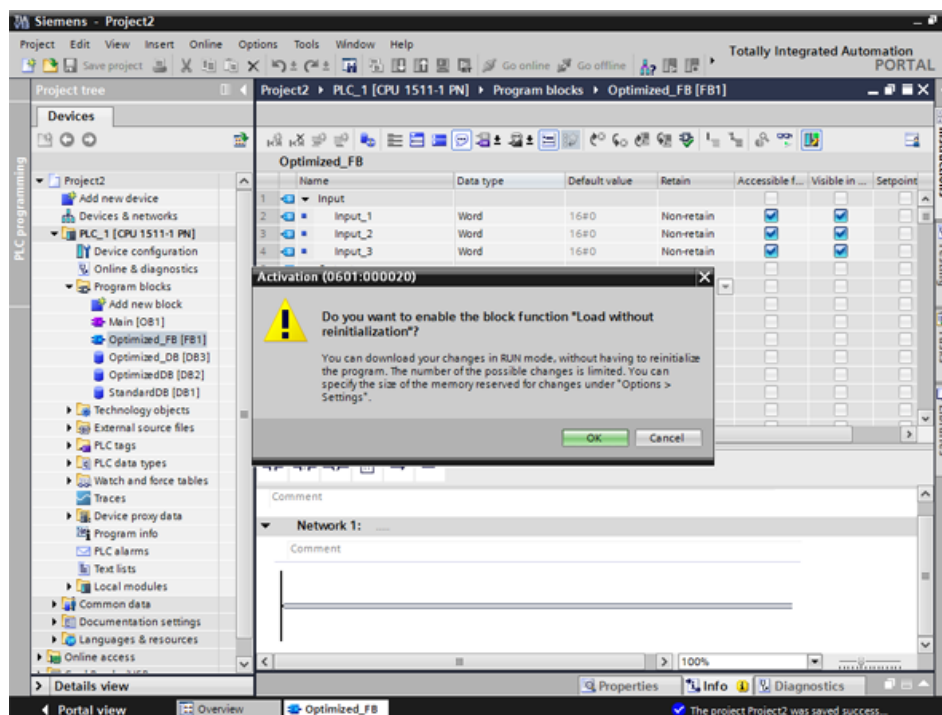
S7-1500 シリーズの CPU の「Optimized データブロック」は、パフォーマンスが最適化されており、形式的な意味においてだけプログラミングされています。最適化データブロックを使用することによって、宣言されたタグがシンボル名を与えられ、固定アドレスを持たないため、プログラムがより効率的に作成できます。

個々のタグの物理的な配置に注意を払わずに、どんな構造を持つデータブロックも作成できます。最適化されたデータへの迅速なアクセスは、システムによってデータストレージが最適化され、管理されているために常に利用可能です。

データタイプの変更は、標準ブロックのエラーのリスクを増加させます。最適化ブロックでは、変更はデータストレージの再構成になります。アドレス指定は固有のままです。

CPU で既に実行されているユーザープログラムを後から編集できるようにするために、S7-1500 CPU はランタイム時のファンクションまたはデータブロックのインターフェースの拡張オプションをサポートしています。CPU を STOP に設定せずに、既にロードされているタグの現在値に影響を与えずに、変更されたブロックをダウンロードできます。

さらに、データブロック自体を定義できます。CPU の値を HMI デバイス用に読み込み専用([Visible in HMI])、または書き込み可能([Accessible from HMI])に定義できます。



3.2.8.2 最適化[Filling]データブロックの拡張と再ロード

概要

以下のセクションで、[Filling]データブロックに最後の充填の日付と時刻を追加し、データブロックを再ロードします。これを行うには、日付と時刻を記録するためのブロックを作成し、ファンクション[Download without reinitialization]を有効にします。

注記: [Download without reinitialization]ファンクションは、CPU へのダウンロード中に、データブロックの実パラメータが上書きされないように保護します。

シンボリックアドレス指定の利点: プロジェクト全体で汎用的に適用された有効なシンボルを使用すると、プログラムコードが読みやすく、理解しやすくなります。これには、以下のような利点があります。

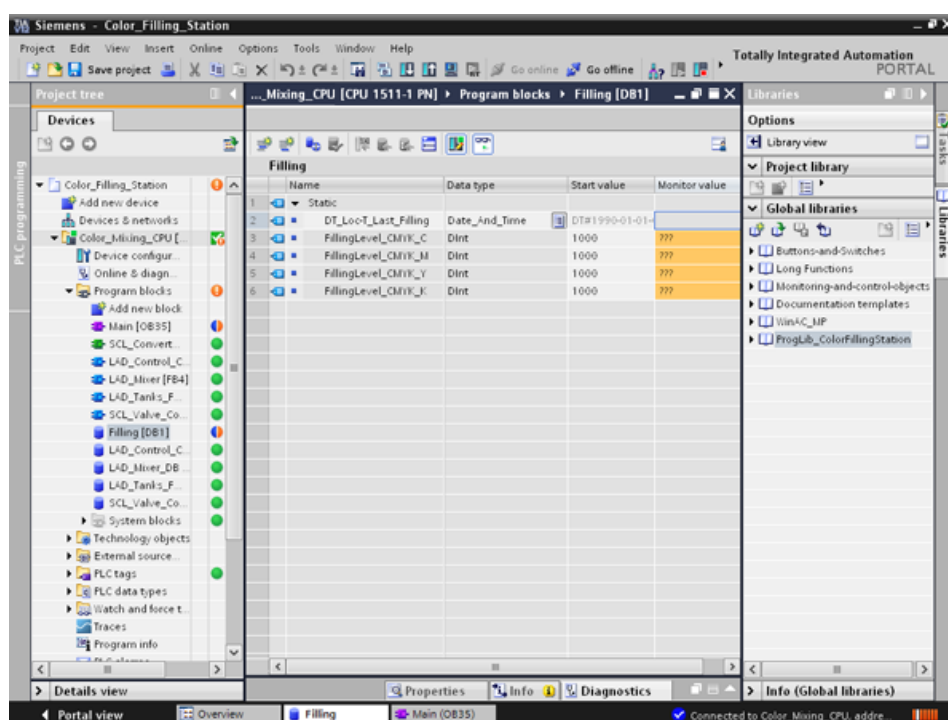
- 詳細なコメントを書く必要がありません。
- データアクセスがより高速です。
- データアクセスの際にエラーが発生しません。
- 絶対アドレスで作業する必要がありません。
- シンボルのメモリアドレスへの割り当ては STEP 7 でモニタされます。これは、使用されるすべてのポイントが、タグの名前やアドレスが変化するとき自動的に更新されることを意味します。

必要条件

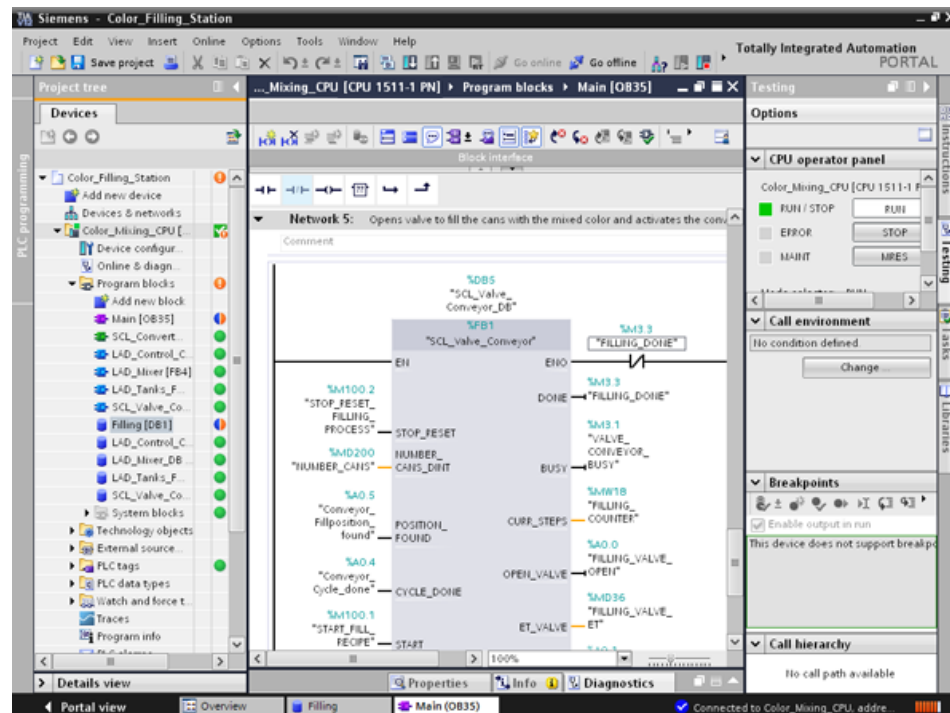
- ライブラリがロードされていること
- プロジェクトがコンパイルされ、CPU にダウンロードされていること

手順

1. [Filling]データブロックを開き、[Main]プログラムブロックを開きます。
2. [Main]プログラムブロック用に[Monitoring on/off]ファンクションを有効にします。
3. [Main]プログラムブロックで、右クリックでネットワーク 3 の[FILLING'
FillingLevel_CMYK_C] I/O のショートカットメニューを開き、[Modify] → [Modify
operand]で選択します。
4. 新しい値を入力し、[OK]をクリックします。
5. [Filling]データブロックの[Download without reinitialization]ファンクションと[Monitor all]
ファンクションを有効にします。
6. [DT_Loc-T_Last_Filling]という名前の新たなパラメータを作成し、データタイプとして
[Date_And_Time]を選択します。



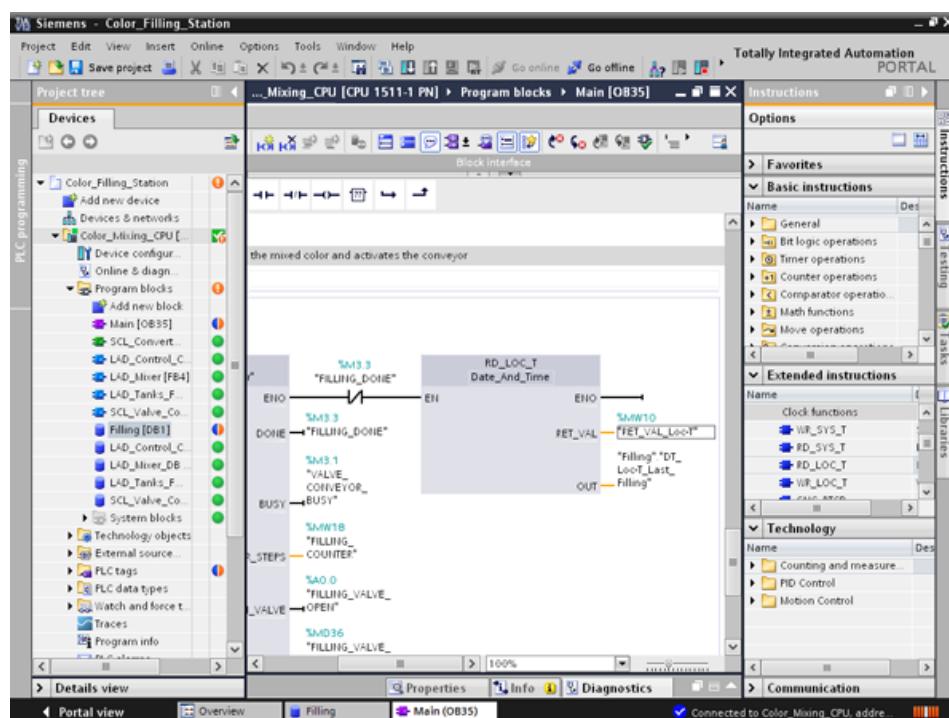
7. ネットワーク 5 の[Main]プログラムブロックにノーマルクローズを挿入し、[FILLING_DONE]パラメータと相互接続します。



8. [Instructions]タブの[Date & time]フォルダを開き、[RD_Loc_T]ブロックを[Main]プログラムブロックに挿入します。

3.2 プログラムの作成

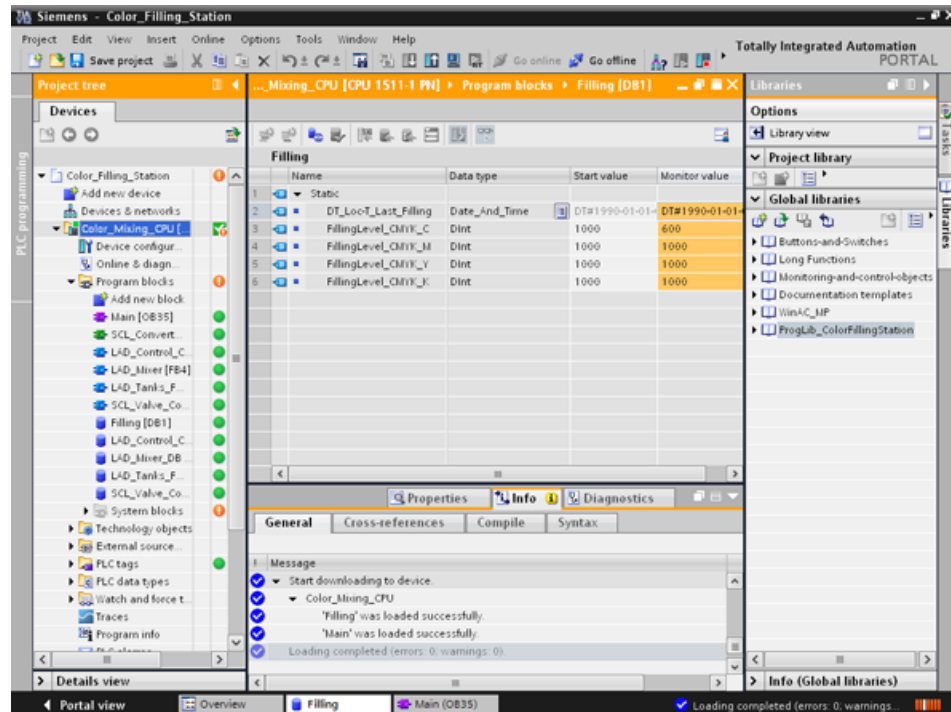
9. [OUT]出力を[DT_Loc-T_Last_Filling]パラメータと相互接続し、[RED_VAL]出力を新しく作成した[RED_VAL_Loc-T]パラメータと相互接続します。[RED_VAL_Loc-T]パラメータ用の格納先として[LAD Tanks Filling Process]データブロックを使用します。



- ## 10. プロジェクトのコンパイルとダウンロード

結果

最後の充填の日付と時刻が再ロードされます。[Filling]データブロックの実パラメータは上書きされません。



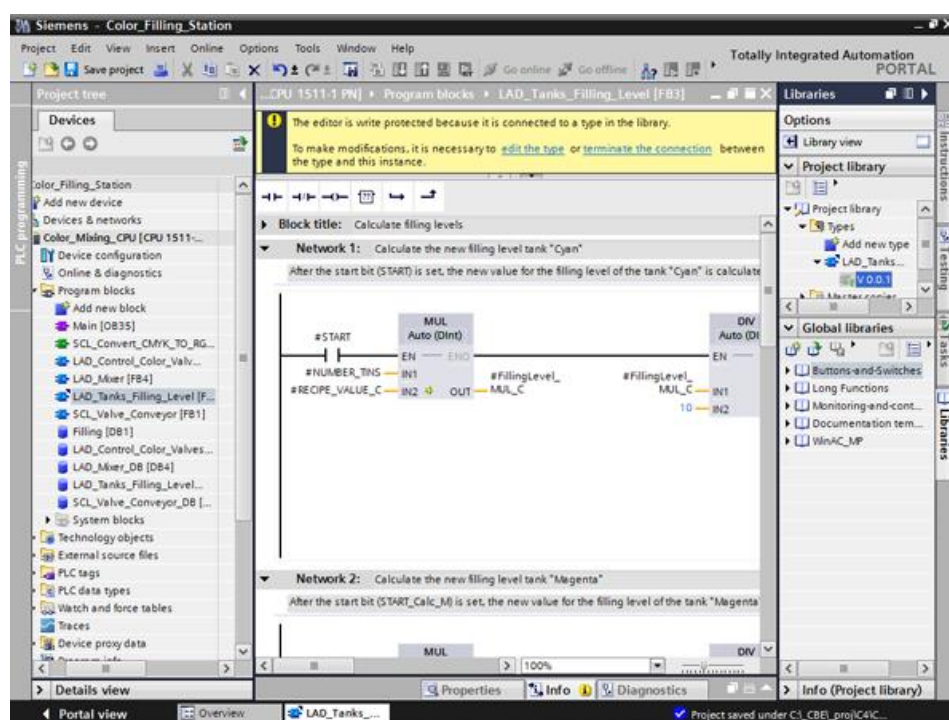
3.2.9 ブロックのバージョン管理

概要

ブロックタイプの使用は、プロジェクトの高度な標準化を保証します。ブロックタイプへのファンクションの拡張部を既存のプロジェクトに容易に統合できます。変更の追跡は、バージョン管理によって確保されます。この例では、プロジェクトライブラリにタイプとして[LAD_Tanks_Filling]ブロックを作成します。ファンクションの拡張として、レベル計算のための 3 つの命令をすべての演算ファンクションを実行する CalculateBox に置き換えます。この最適化は、一時的なタグの必要が少なくなり、さまざまなプログラミング言語によるブロック間の切り替えがもはや必要なくなることを意味します。

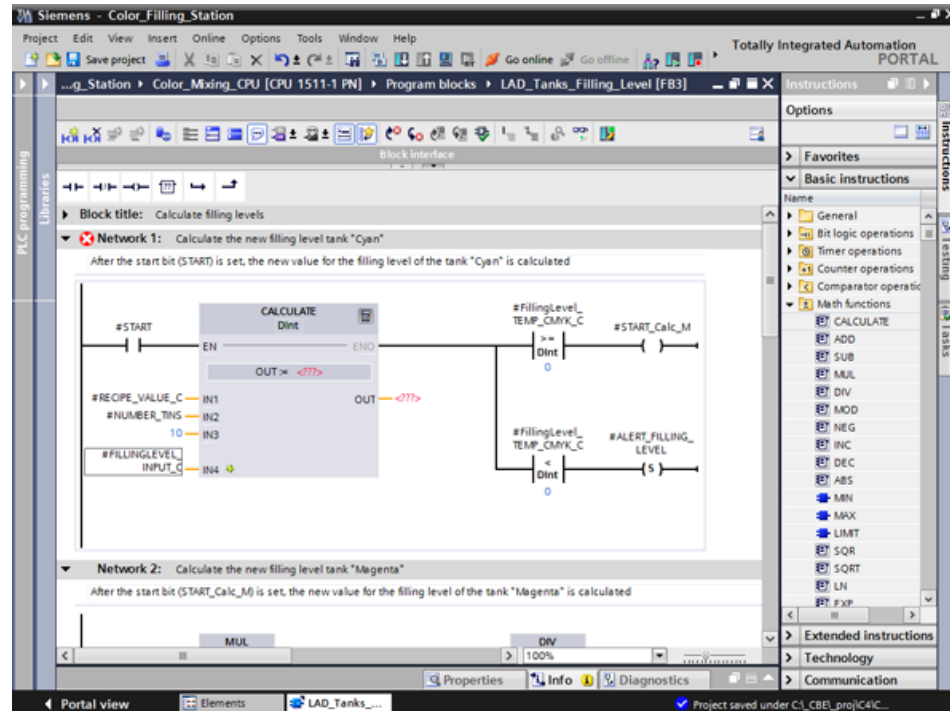
手順

1. [LAD_Tanks_Filling]ブロックをコンパイルし、次にそれを[Types]の下プロジェクトライブラリに挿入します。
2. [Edit type]の新しいブロックバージョンを作成します。

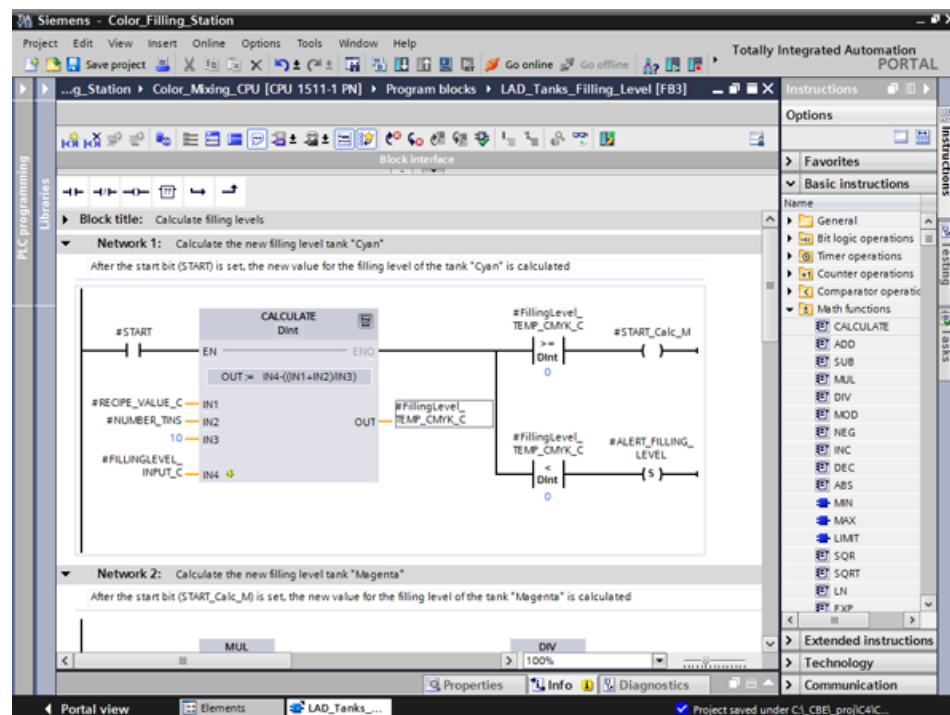


3. [Basic instructions] → [Mathematical functions]ライブラリから CALCULATE 命令を挿入します。
4. ブロックから MUL、DIV および SUB 命令を削除します。

5. 2つの入力を CALCULATE 命令に挿入し、入力を相互接続します。



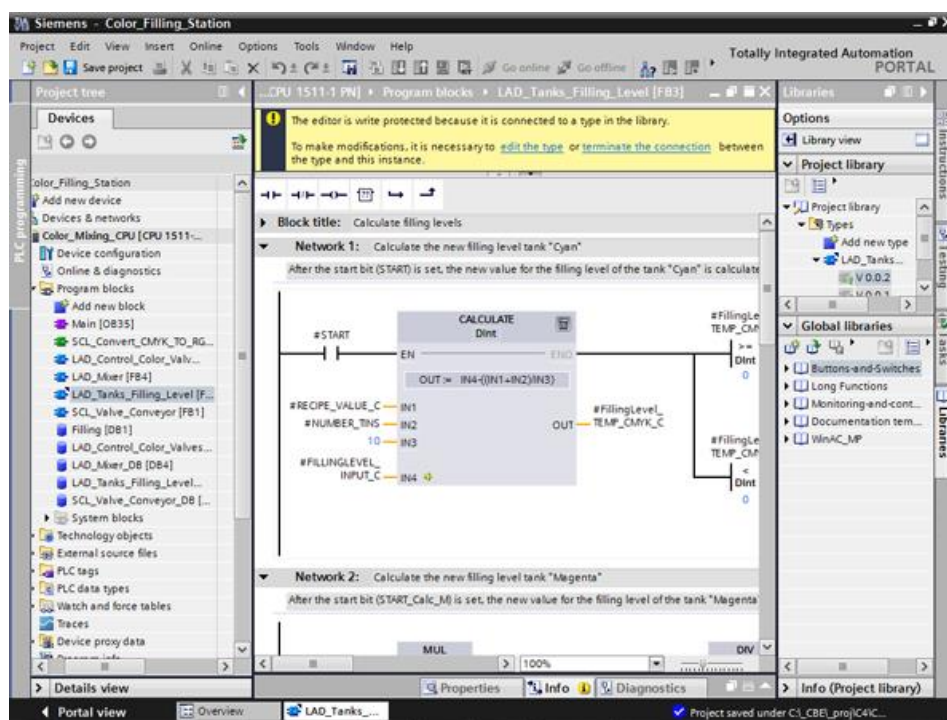
6. 計算式を定義し、次に出力を相互接続します。



7. ブロックバージョンをリリースします。

結果

ブロックタイプの改訂版が新しいバージョン番号でライブラリに保存されます。



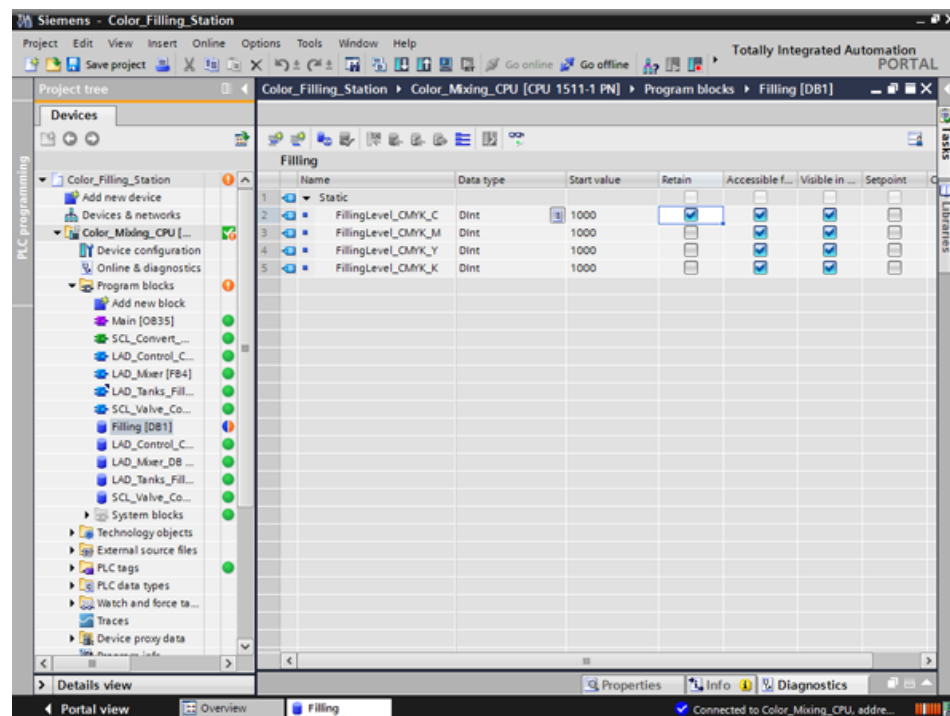
3.2.10 保持設定

概要

すべてのタグは、電源オフの後などの CPU の起動中に、設定された開始値で初期化されます。遮断の直前にタグが保持していた最新の値が、初期値で上書きされます。これを防ぐために、タグを保持型と定義します。保持型のタグは、再起動後でも値を保持します。この例では、塗料貯蔵タンクのレベルが CPU の保持メモリ領域でバックアップされます。

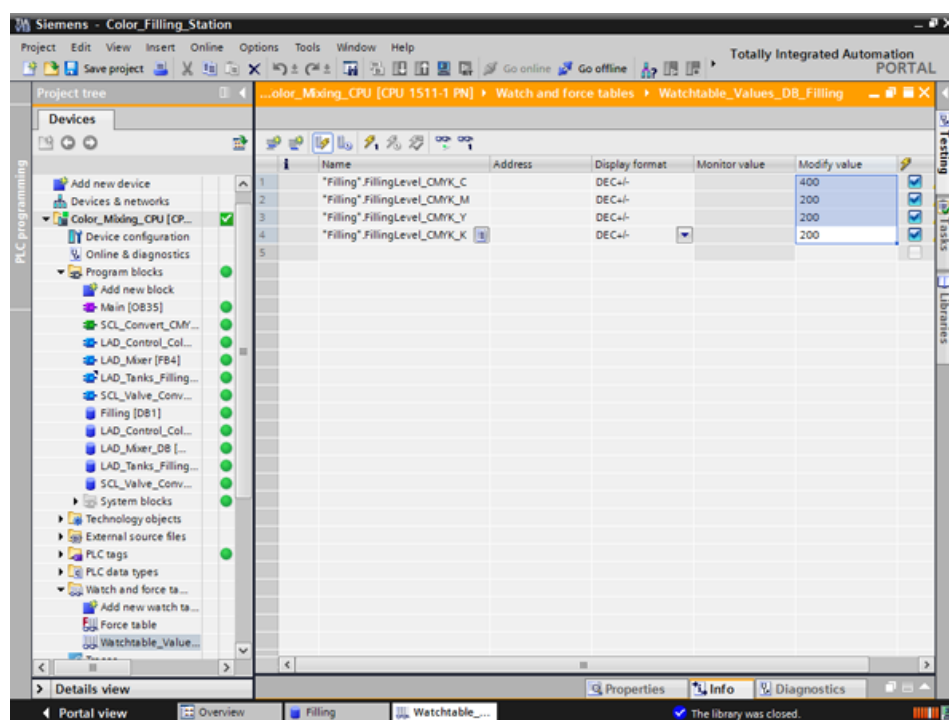
手順

1. CPU オンラインに接続します。
2. [Filling]データブロックの[Cyan]エントリの[Retain]チェックボックスを有効にします。



3. 変更を CPU にダウンロードします。
4. [Watchtable]オブジェクトをライブラリからプロジェクトにドラッグします。このオブジェクトは、制御値に含まれている占有レベルタグを含みます。

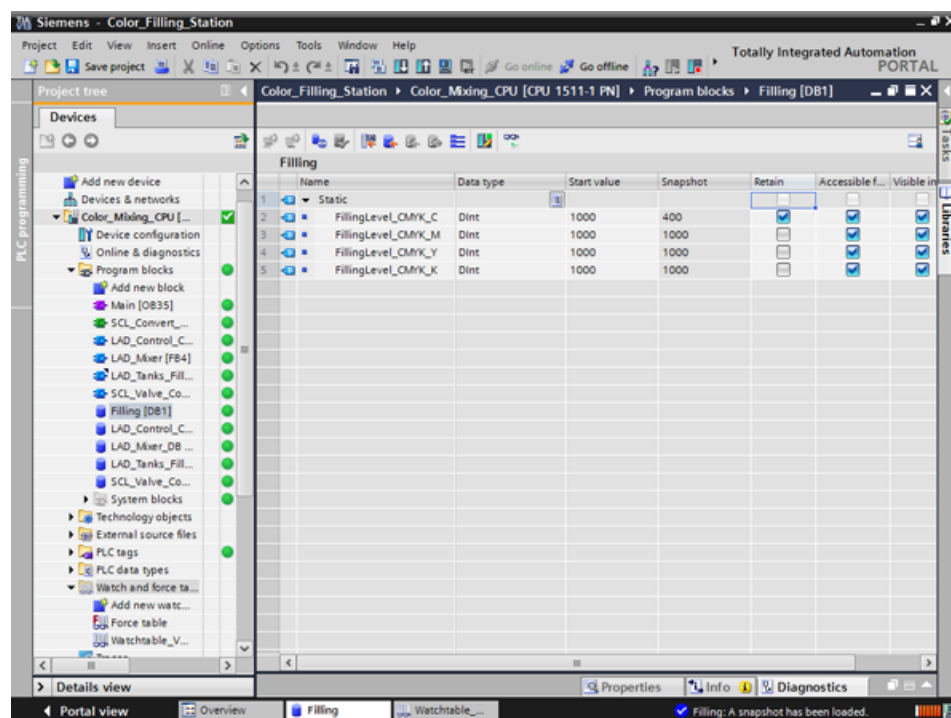
5. [Modify now]を使用して制御値を CPU に転送します。



6. CPU へのオンライン接続を閉じます。電源オフをシミュレートするために、CPU への電源接続を解除します。
7. 電源を再接続し、CPU にオンラインでアクセスします。[Filling] DB に対して[Monitor all]を有効にします。

結果

[Cyan]の占有レベルは、保持メモリ領域から読み込まれます。他のすべての占有レベルは、開始値で再度初期化されます。



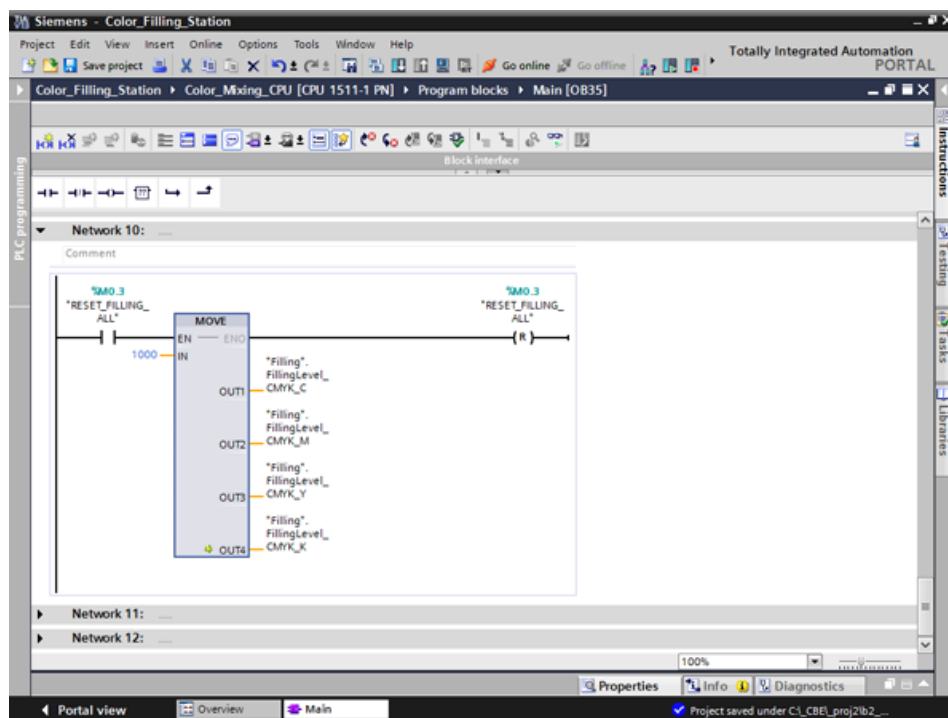
3.2.11 EN/ENO メカニズムの有効化

概要

各種命令の EN/ENO メカニズムにより、ランタイムエラーの検出や、プログラムクラッシュの回避が可能になります。新たに挿入された ENO 命令は、デフォルトで無効になっています。次に、ENO 許可出力を有効にできます。すべての塗料貯蔵タンクの占有レベルが同時に開始値(1000)にリセットされる、新しいネットワークでこれを使用できます。

手順

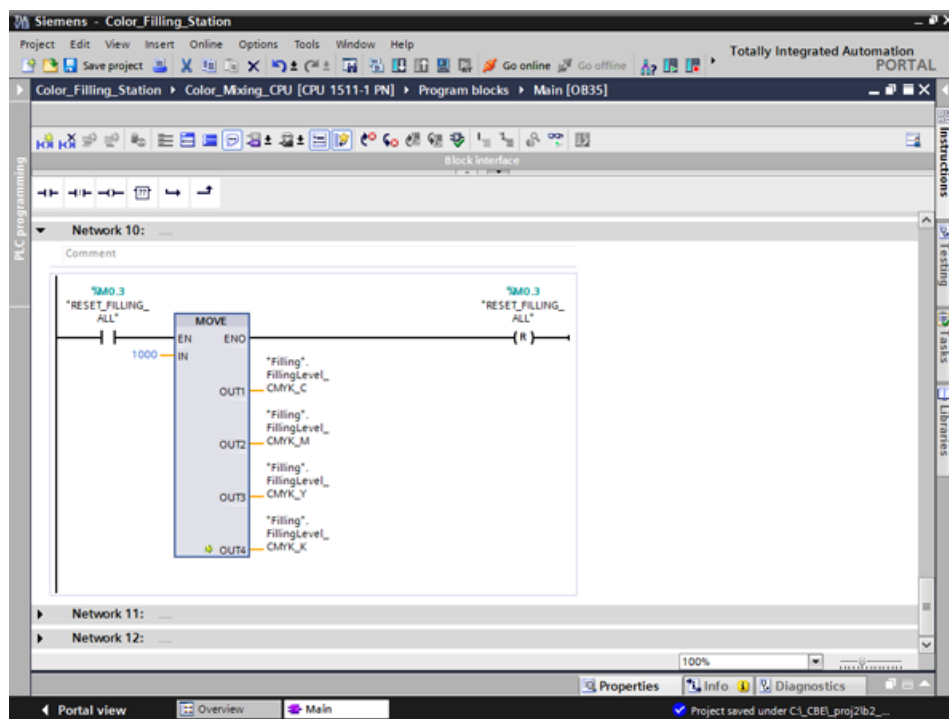
1. Main[OB35]プログラムブロックを開き、MOVE 命令をネットワーク 10 に挿入します。
2. 命令を合計で 4 出力に拡張します。
3. MOVE 命令の前にノーマルオープン挿入します。
4. MOVE 命令の後にリセットコイルを挿入します。
5. MOVE 命令の入出力を相互接続します。



6. ENO ショートカットメニューで命令を生成します。

結果

EN/ENO メカニズムが、このブロックのために相互接続されます。実行中にエラーがない場合、ENO 許可出力はシグナル状態「1」です。実行中にエラーがある場合、ENO 許可出力はシグナル状態「0」です。



3.2.12 コメントファンクションの使用

概要

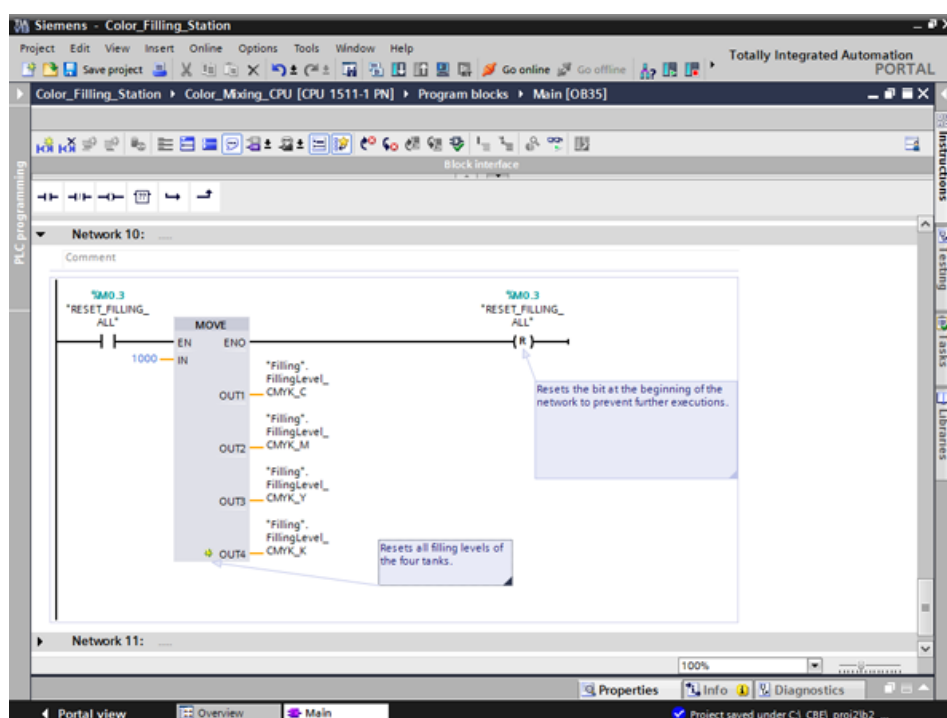
MOVE と Reset 命令は、詳細なコメントを付けるべきです。

手順

1. ショートカットメニューを使用してコメントを挿入します。
2. コメントテキストを入力します。

結果

命令とコイルのコメントが入力されます。

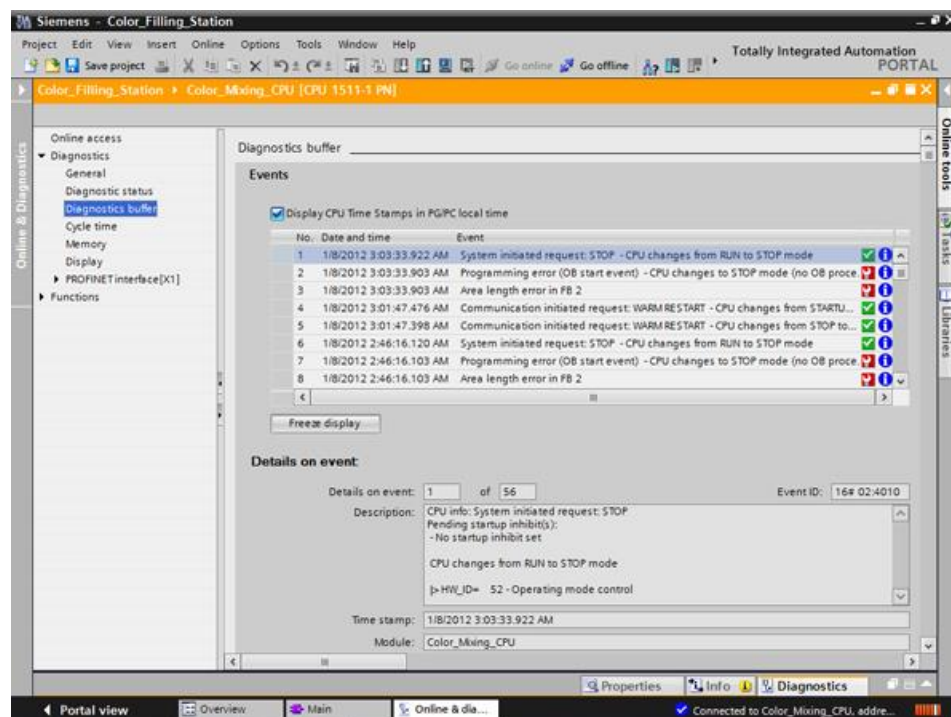


3.2.13 ローカルエラーハンドリング

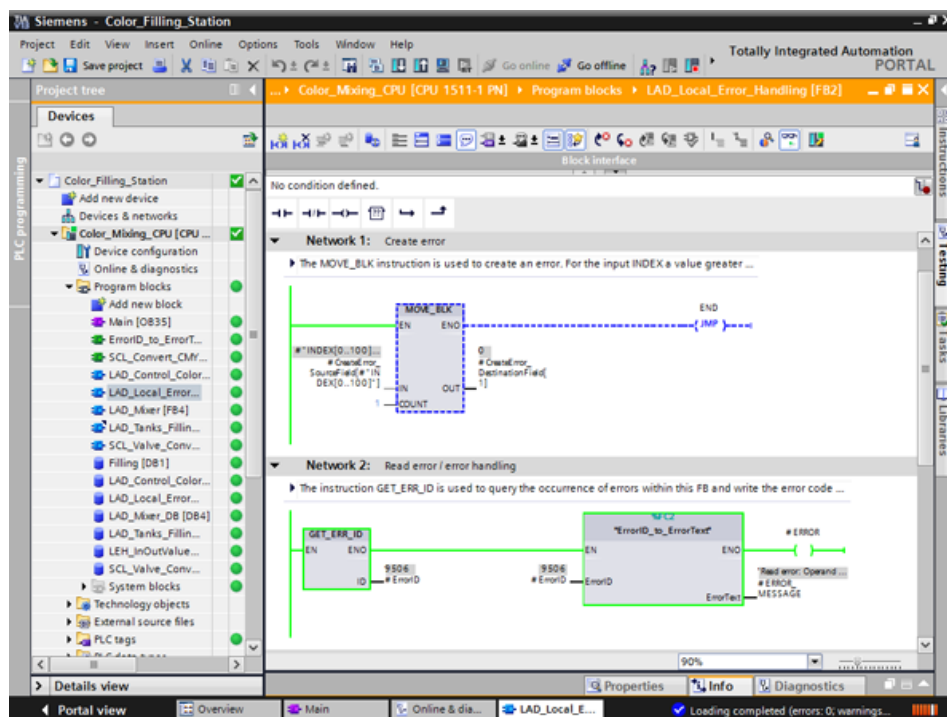
3.2.13.1 ブロック内のエラー処理

手順

S7-300/400 の CPU と異なり、S7-1500 の CPU ははるかに少ない頻度ですがエラーで STOP 状態に移行します。エラーが発生する場合、CPU の診断バッファに入力されます。各ブロックのローカルエラーハンドリングを使用して CPU STOP を回避します。できれば、ユーザープログラムの開発中にローカルエラーハンドリングを有効にすべきです。



情報を正確に評価できます。たとえば、STL/FBD/LAD および SCL プログラムでブロックにエラー処理をプログラムします。ブロックは、[GET_ERROR_ID]命令によって評価されるエラーID を生成します。MAIN ブロックとファンクションブロックの両方で、[GET_ERROR_ID]命令を呼び出すことができます。CPU は RUN モードのままです。



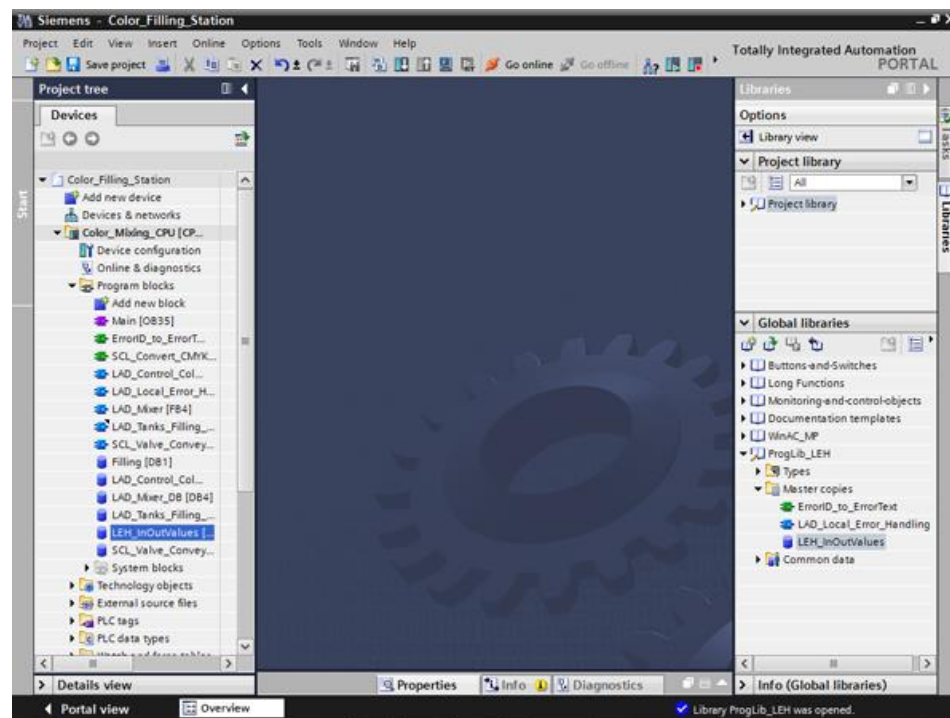
3.2.13.2 ローカルエラーハンドリング用のブロックのダウンロード

概要

ローカルエラーハンドリングを説明するために、プロジェクトに[ProgLib_LEH]ライブラリのブロックをロードします。ブロックはローカルエラーハンドリングを説明するためにのみ使用され、プロジェクトでは使用されません。

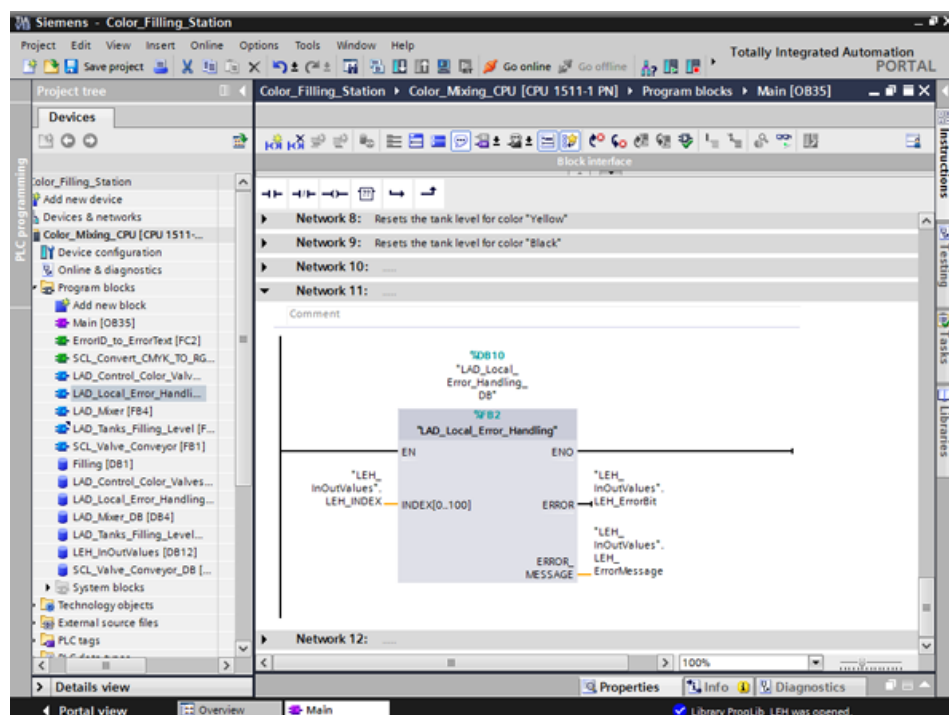
手順

1. グローバルライブラリ[ProgLib_LEH]を開きます。
2. マスタコピーからプロジェクトにブロックをコピーします。



3. [Main]ブロックの空のネットワークに[LAD_Local_Error_Handling]ファンクションブロックを呼び出します。

4. [LAD_Local_Error_Handling]ファンクションブロックのパラメータを[LEH_InOutValues]データブロックのタグと相互接続します。



5. CPU オンラインに接続します。
6. 変更をコンパイルし、CPUにダウンロードします。

結果

[INDEX[0..100]]入力パラメータで[LEH_INDEX]タグを使用して、次のプログラミングエラーをトリガします。たとえば、入力パラメータを[101]にセットした場合、出力パラメータでエラーが報告されます。

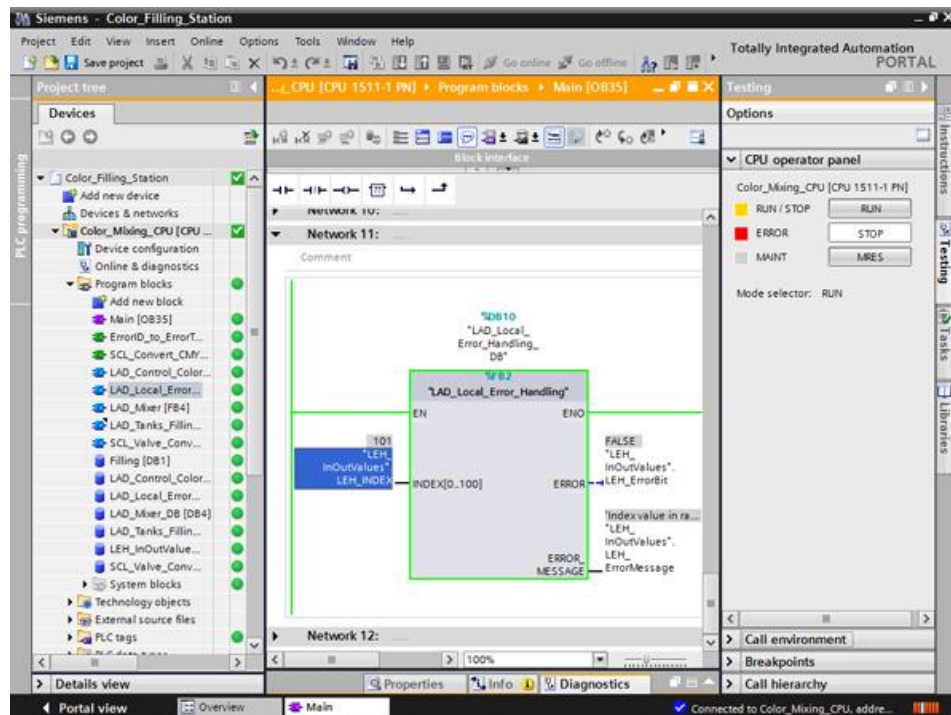
3.2.13.3 ローカルエラーハンドリングなしでエラーを生成

概要

以下の手順を実施して、ローカルエラーハンドリングを使用しないで、または対応する OB を作成せずにプログラミングエラーをトリガします。

手順

1. [Monitor]ファンクションを有効にします。
2. [LEH_INDEX]タグの値を[101]などの無効な値にセットします。[Testing]ダイアログで、ERROR LED が短い時間点滅し、CPU が RUN から STOP に移行します。



3. 診断バッファに切り替えます。エラーおよびエラー応答が診断バッファに表示されます。
4. CPU を RUN にセットし直します。

結果

STOP から RUN への移行は、[LEH_INDEX]タグを開始値[0]にリセットします。
これで、自動的に問題が解決します。

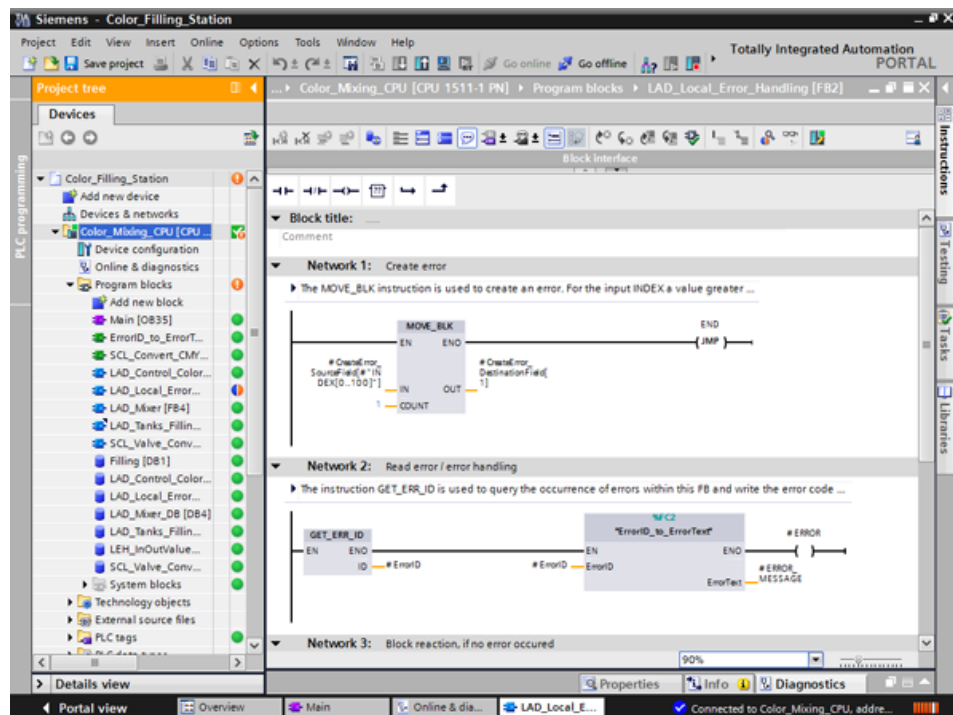
3.2.13.4 ローカルエラーハンドリングを使用してエラーを生成

概要

ローカルエラーハンドリングがエラーにエラーメッセージ付きで応答するために、[GET_ERR_ID]命令と ENO ビットを使用する以下の手順を実行してください。これは、CPU が RUN モードのままになることを意味します。

手順

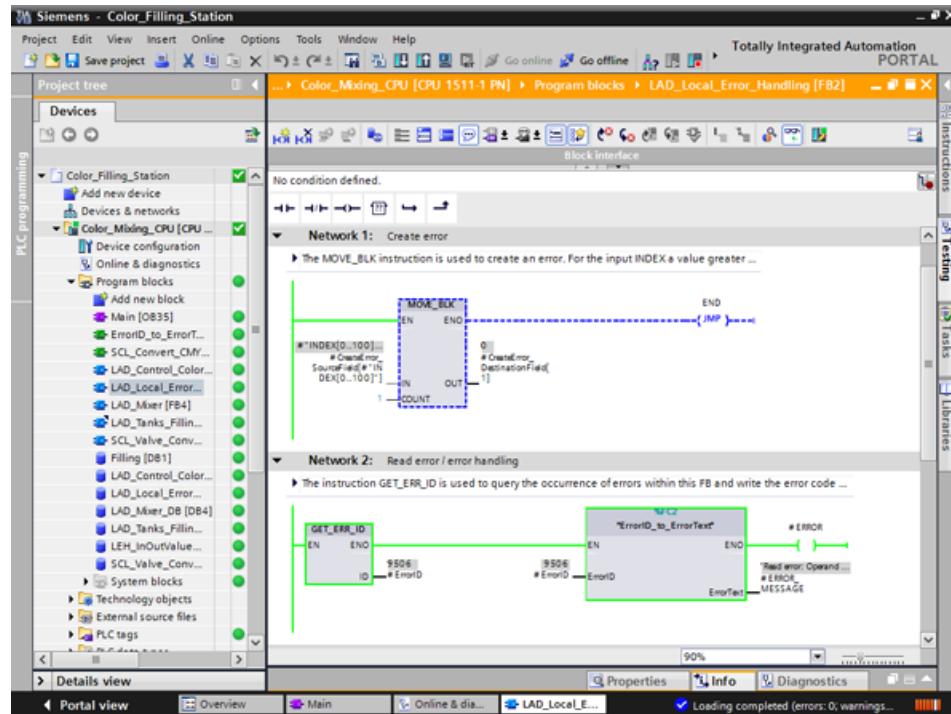
1. [LAD_Local_Error_Handling]ファンクションブロックを開きます。
2. [GET_ERR_ID]命令を 2 番目のネットワークに挿入し、[ID]出力と相互接続します。



3. プロジェクトツリーから[ErrorID_to_ErrorText]ファンクションを呼び出します。
4. [ErrorID_to_ErrorText]ファンクションのパラメータを相互接続し、エラーコードをエラーメッセージに変換できるようにします。
5. 変更を CPU にダウンロードします。
6. 無効な値[101]などを入力して[Main]オーガニゼーションブロックのエラーをトリガします。エラーメッセージが、[ERROR_MESSAGE]パラメータに出力されます。

結果

エラーメッセージは、エラーが修正されない限り出力されます。エラーを修正するには、[LEH_INDEX]タグに有効値を割り当てるか、CPU を再起動します。



3.3 ビジュアライゼーションの設定

3.3.1 現在のサンプルプロジェクト

アプリケーション用のサンプルプロジェクト

TIA ポータルを使用してカラーミキシングシステムを構成するには、サンプルプロジェクト「Color_Filling_Station」を作成します。次のプロジェクトコンポーネントが、既にサンプルプロジェクト用に存在します。CPU ユーザープログラムのプログラムブロックとタグテーブル、および必要な HMI 画面、HMI タグ、およびスクリプト付きの構成されたコンフォートパネルが存在します。

このセクションでは、サンプルプロジェクトの個々のプロジェクトコンポーネントの間の関係を説明します。後で、必要なコンフィグレーション手順を自分で実際に行うことになります。

3.3.2 HMI 構成

3.3.2.1 概要

HMI 構成の紹介

支給されたプロジェクトは、[global library]のプログラムされた CPU および 構成済み HMI デバイスを含みます。

HMI 構成

このセクションで、HMI デバイスと HMI 構成を紹介します。

追加情報

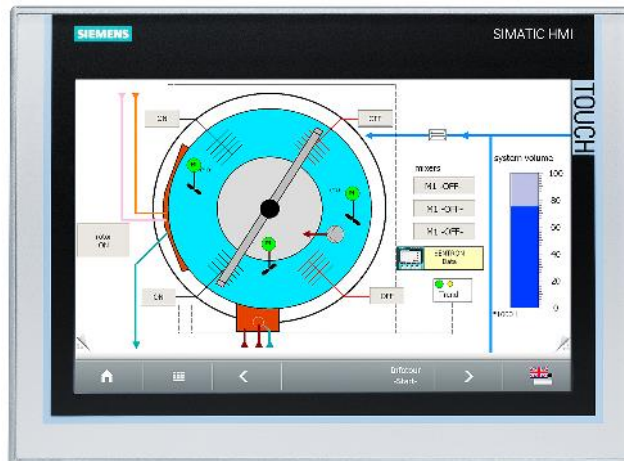
HMI 構成について、詳しくは下記を参照してください。

『Getting Started WinCC V13 Comfort Panels, Runtime Advanced』

3.3.2.2 SIMATIC HMI コンフォートパネル

SIMATIC HMI コンフォートパネル

コンフォートパネルシリーズの TP1200 コンフォート HMI デバイスは、カラーミキシングシステムの操作に使用されます。



コンフォートパネルは、特に PROFINET および PROFIBUS 環境での困難な HMI タスクに最適で、以下の機能を特徴としています。

- 高品質のハウジングと各種インターフェース
- 産業用ワイドスクリーンディスプレイ(大型可視化領域、安定した最適視野角と最大輝度)
- 横または縦の取り付けが可能
- システム診断ビューによる正確な診断

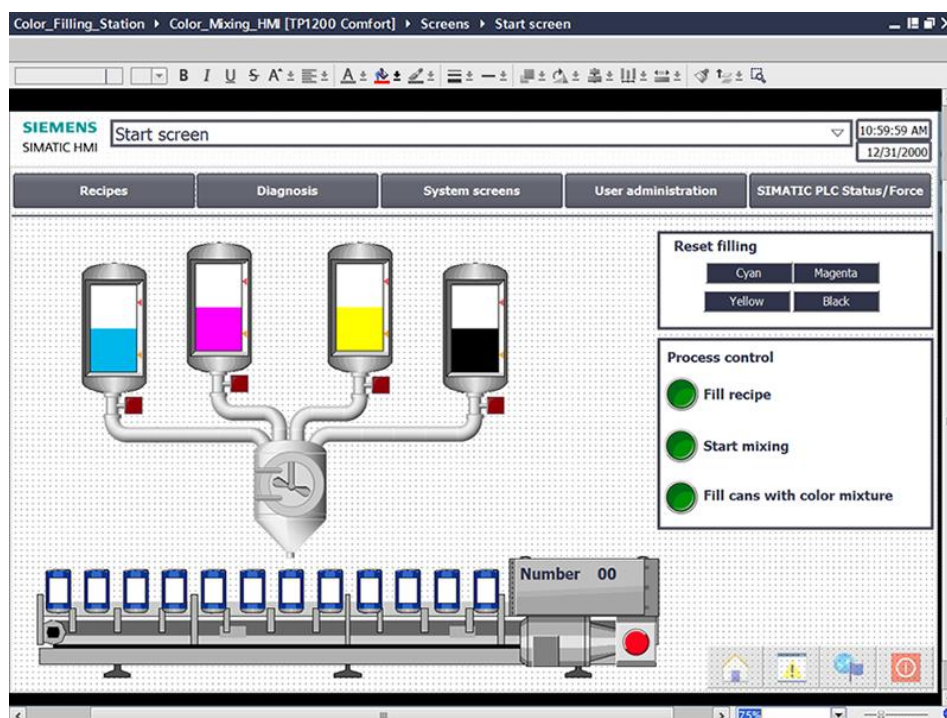
3.3.2.3 HMI 画面

HMI 画面

それぞれの HMI デバイスにロードされた画面を使用して、機械やプラントをランタイム時に操作し、モニタします。

プロジェクトナビゲーションの[Screens]の WinCC で各画面を管理します。

HMI デバイスの開始画面は、カラーミキシングシステム、および最も重要なステータス情報や主要な数値を可視化するのに設計されています。



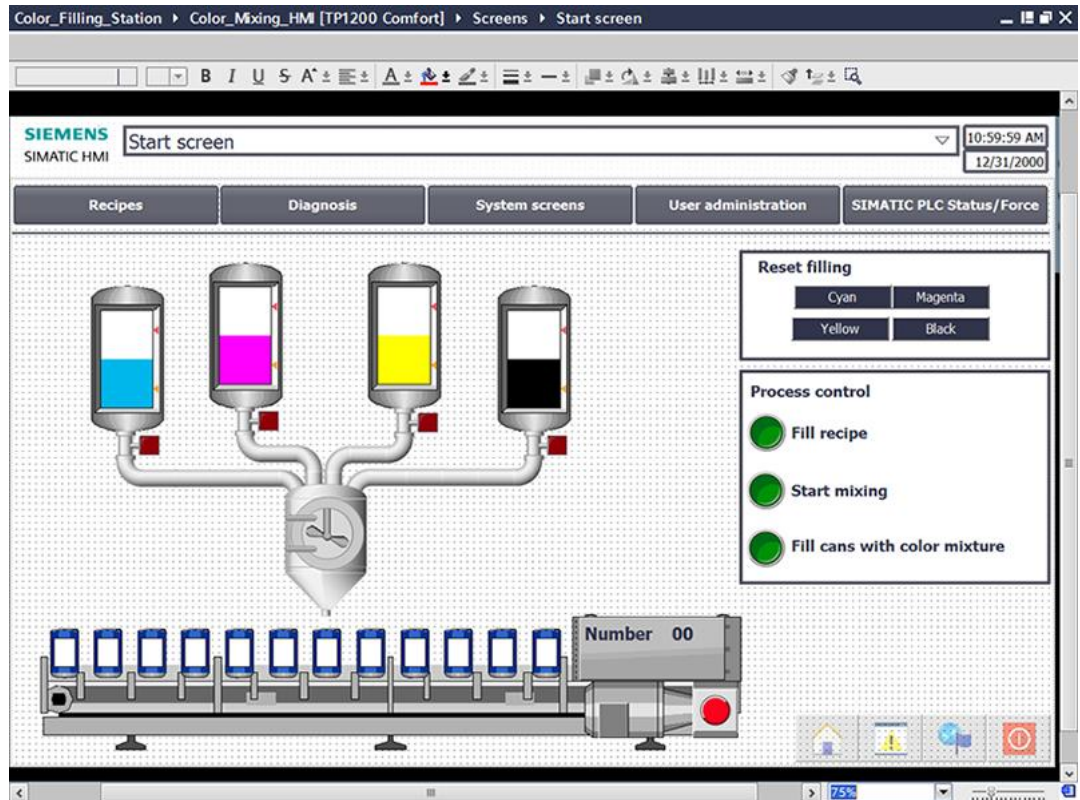
カラーミキシングシステムは、以下のエレメントを含みます。

- 占有レベルが表示されている各プリントカラーのカラータンク
- ミキサー
- ミキサーへの配管
- 非常停止スイッチ付きコンベヤベルト

3.3.2.4 追加のコントロールエレメント

追加のコントロールエレメント

プロセスステップ[Mixing color]と[Filling color]は、動的なビジュアライゼーションオブジェクトを使用したアニメーションで表示されます。



プロジェクト例の開始画面は、追加のコントロールオブジェクトを含みます。

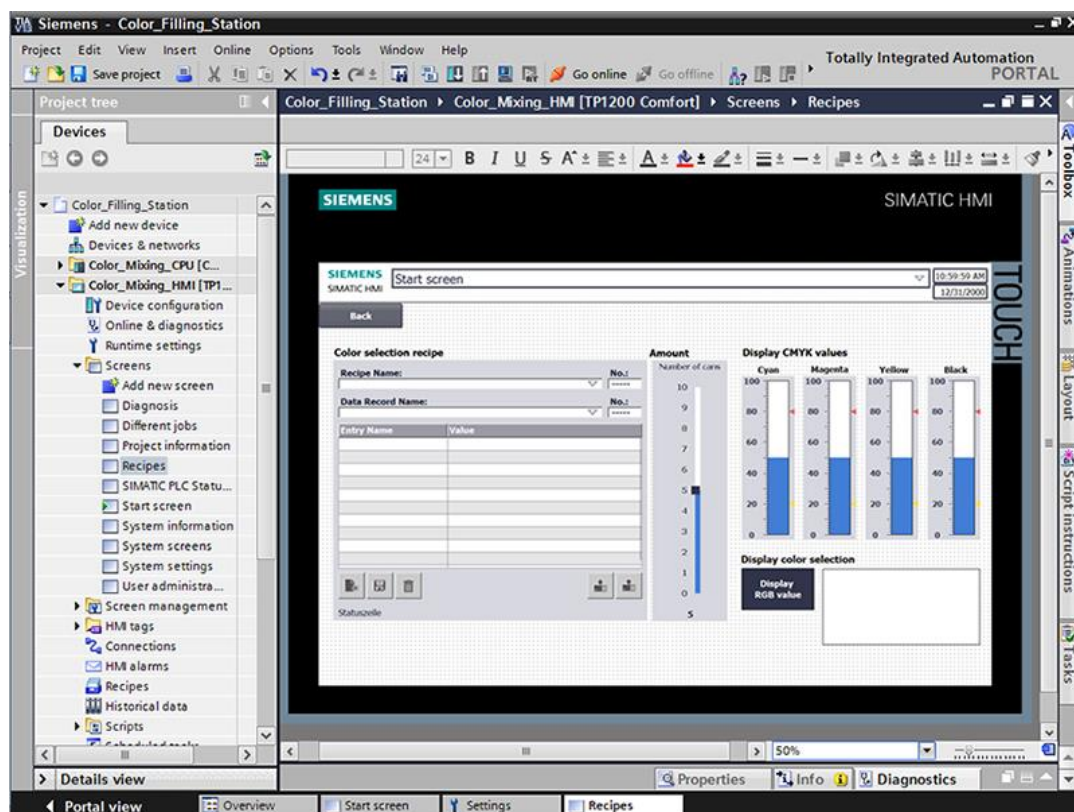
- 画面変更用ボタン
- 占有レベルをリセットするボタン
- システムの操作とモニタ用のボタン: 充填レシピ、ミキシングプロセスの開始、混色の充填

3.3.2.5 レシピ

レシピ

レシピは、混合比などに関連する生産パラメータを扱います。

必要な混合比は、1つの操作で HMI デバイスからカラーミキシングシステムへ転送できます。たとえば、濃いオレンジ色からシグナルイエローへの生産の切り替えが一回でできます。



カラーミキシングシステムは、混合された色[Orange]、[Amber]、[Green]、および[Red]を生産できます。

レシピデータレコードは、各カラー用に作成されます。レシピデータレコードは、それぞれの混合された色になる基本色の割合を含みます。

レシピは関連パラメータとカラーの個別の色調のための混合比が保存されるレシピデータレコードから構成されます。

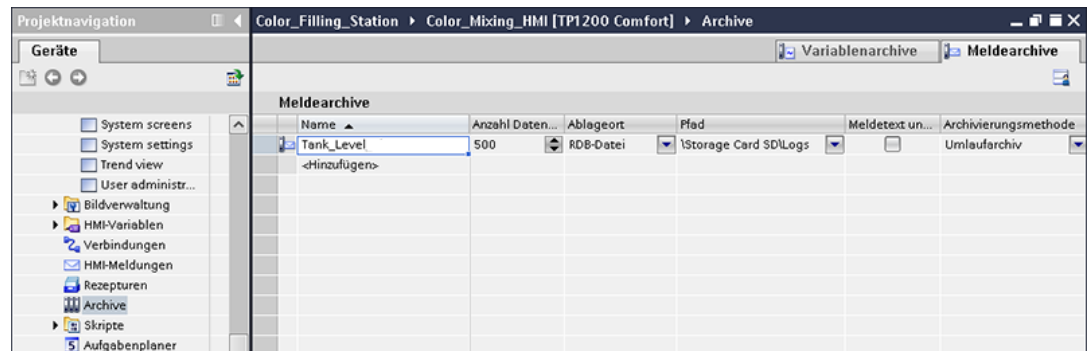
3.3.2.6 アーカイブ

アーカイブ

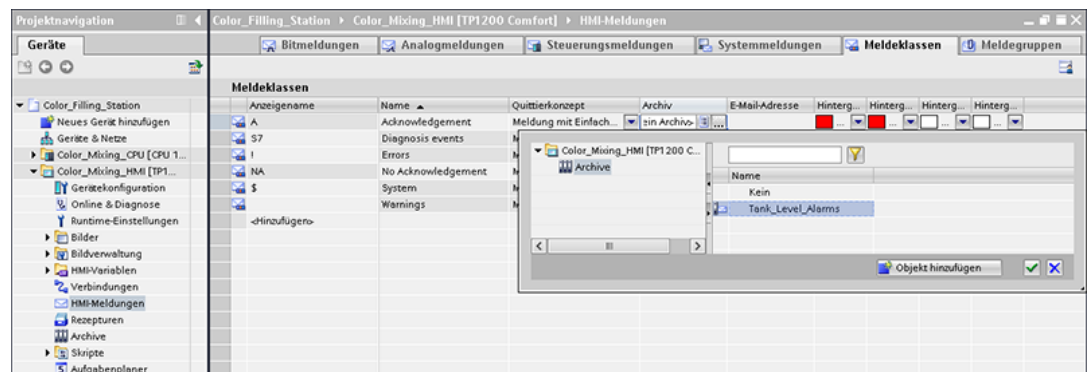
システムの運用上のイベントを記録するために、生産中に生成されたアラームとプロセス値がログに保存されます。

その後、アラームとプロセスデータログを評価できます。

カラータンクの占有レベルは、カラーミキシングシステムのために文書化されます。このために、アラームログ[Tank_Level]を構成しています。



このログは、シフト中に占有レベルが低すぎる場合のアラームと、高すぎる場合のアラームを保存します。



3.3.2.7 ユーザー定義ファンクション

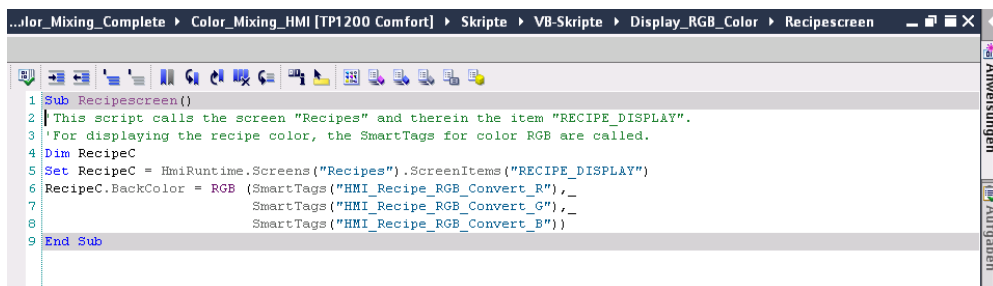
スクリプト

ユーザー定義ファンクションを使用して、HMI デバイス用に追加の機能をプログラムできます。

WinCC は、ユーザー定義ファンクションを作成するために、VBS プログラミングインターフェースが組み込まれています。

プロジェクト例では、2つのユーザー定義ファンクションを使用して、混合したプリントカラーを別々の画面でモニタに表示します。

- [Recipescreen]は、[Recipes]画面に選択した色の四角形を表示します。

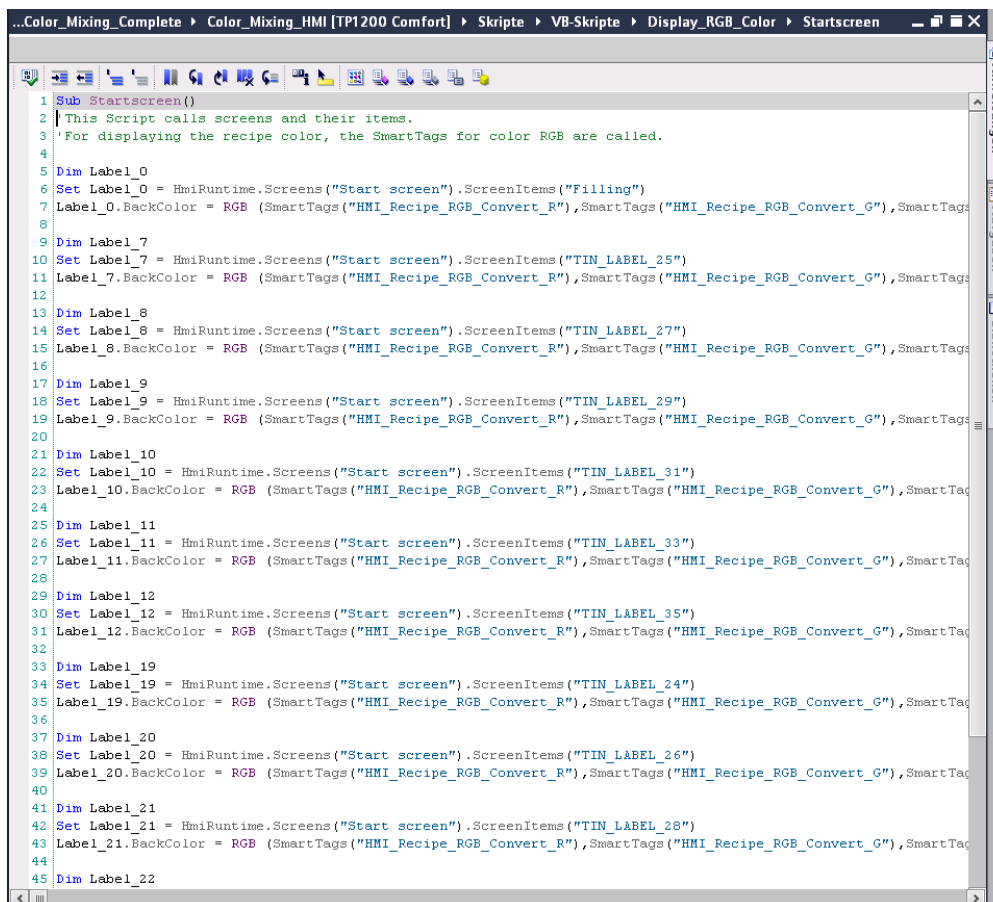


```

1 Sub Recipescreen()
2 | This script calls the screen "Recipes" and therein the item "RECIPE_DISPLAY".
3 | For displaying the recipe color, the SmartTags for color RGB are called.
4 Dim RecipeC
5 Set RecipeC = HmiRuntime.Screens("Recipes").ScreenItems("RECIPE_DISPLAY")
6 RecipeC.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), _
7 | SmartTags("HMI_Recipe_RGB_Convert_G"), _
8 | SmartTags("HMI_Recipe_RGB_Convert_B"))
9 End Sub

```

- [Startscreen]は、開始画面のシステム概要で現在混合されている色で充填された缶のラベルを表示します。



```

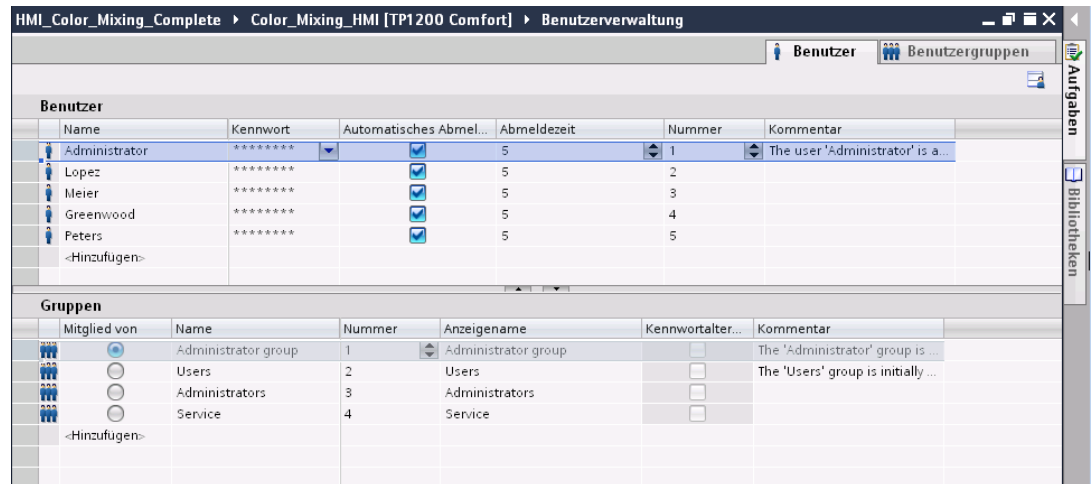
1 Sub Startscreen()
2 | This Script calls screens and their items.
3 | For displaying the recipe color, the SmartTags for color RGB are called.
4
5 Dim Label_0
6 Set Label_0 = HmiRuntime.Screens("Start screen").ScreenItems("Filling")
7 Label_0.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
8
9 Dim Label_7
10 Set Label_7 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_25")
11 Label_7.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
12
13 Dim Label_8
14 Set Label_8 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_27")
15 Label_8.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
16
17 Dim Label_9
18 Set Label_9 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_29")
19 Label_9.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
20
21 Dim Label_10
22 Set Label_10 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_31")
23 Label_10.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
24
25 Dim Label_11
26 Set Label_11 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_33")
27 Label_11.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
28
29 Dim Label_12
30 Set Label_12 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_35")
31 Label_12.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
32
33 Dim Label_19
34 Set Label_19 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_24")
35 Label_19.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
36
37 Dim Label_20
38 Set Label_20 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_26")
39 Label_20.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
40
41 Dim Label_21
42 Set Label_21 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_28")
43 Label_21.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), SmartTags("HMI_Recipe_RGB_Convert_G"), SmartTags("HMI_Recipe_RGB_Convert_B"))
44
45 Dim Label_22

```

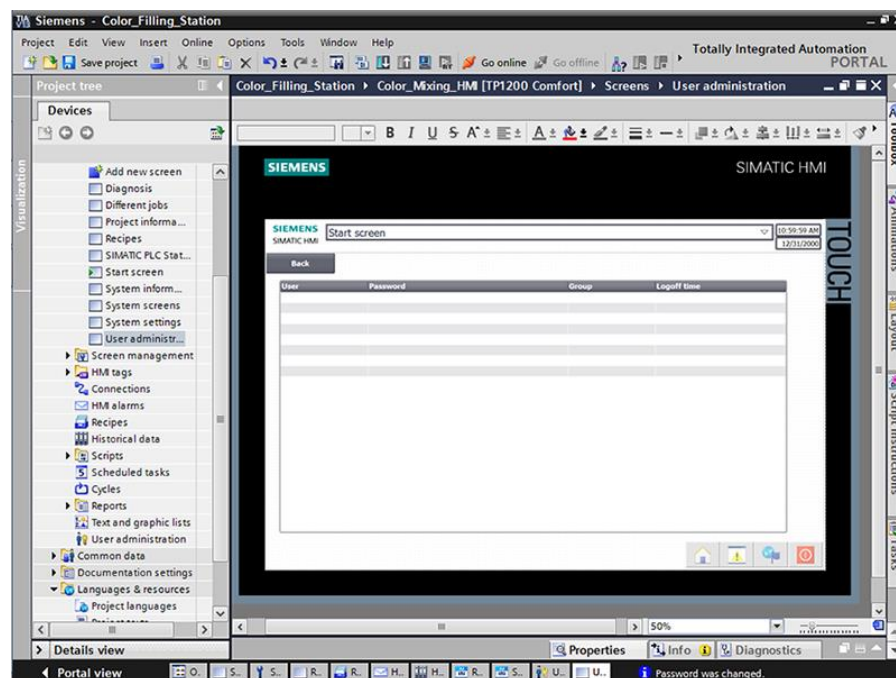
3.3.2.8 ユーザー管理

ユーザー管理

WinCC には、安全関連操作を特定のユーザーグループに制限するオプションがあります。このオプションで、データとファンクションをランタイム時に未承認のアクセスから保護します。



[User view]オブジェクトは、HMI デバイスのユーザーとパスワードの管理が設定できます。



ユーザー管理権限を持つユーザーは、ユーザービューですべてのファンクションへのアクセス権を持ちます。

ユーザーを作成、削除でき、自分のパスワードや、他のユーザーのパスワードを変更できます。

3.3.2.9 多言語の使用

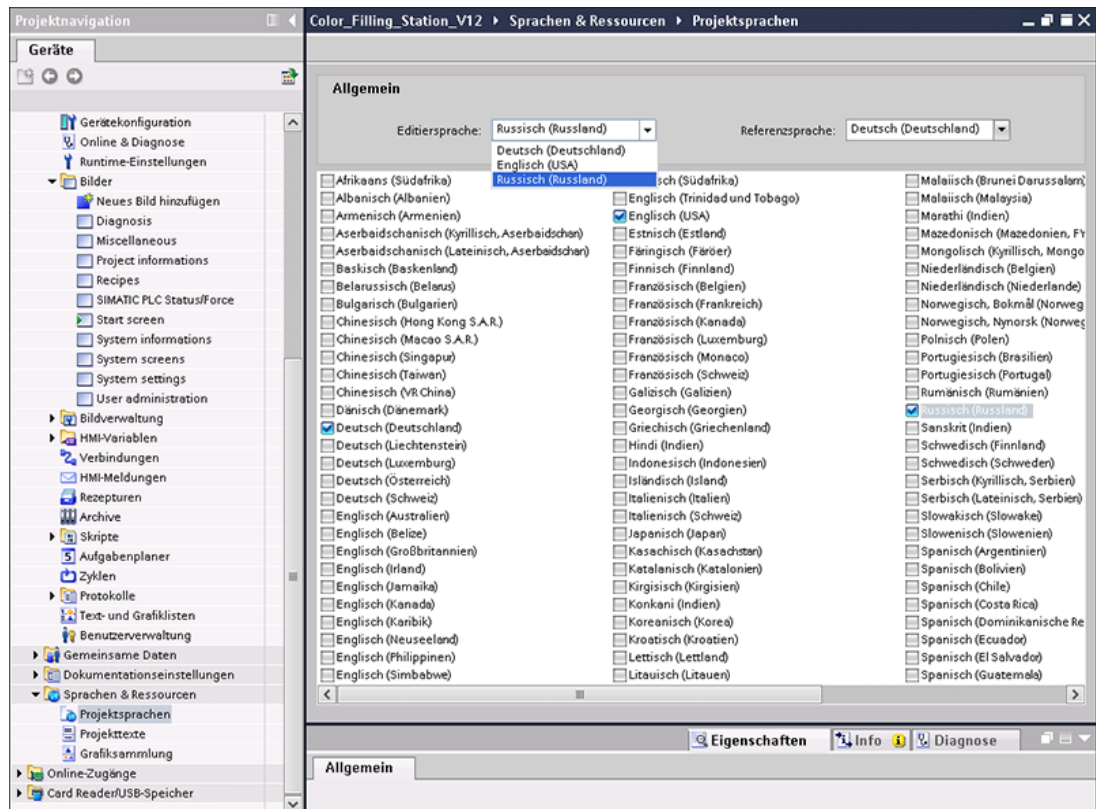
多言語の使用

WinCC は、多言語ユーザーインターフェースをサポートします。

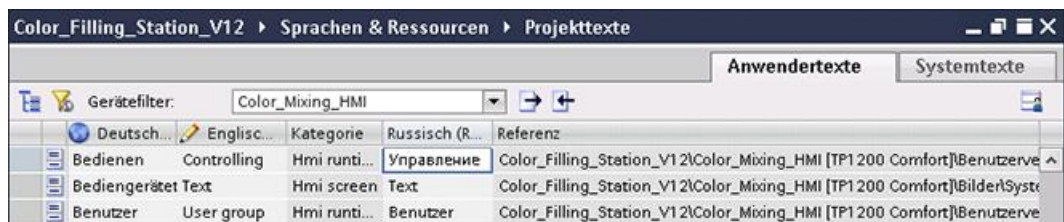
カラーミキシングシステムは、ロシアにある新設の子会社で運営されています。

ロシア語のユーザーインターフェースは、メンテナンスと技術サービスのために必要です。

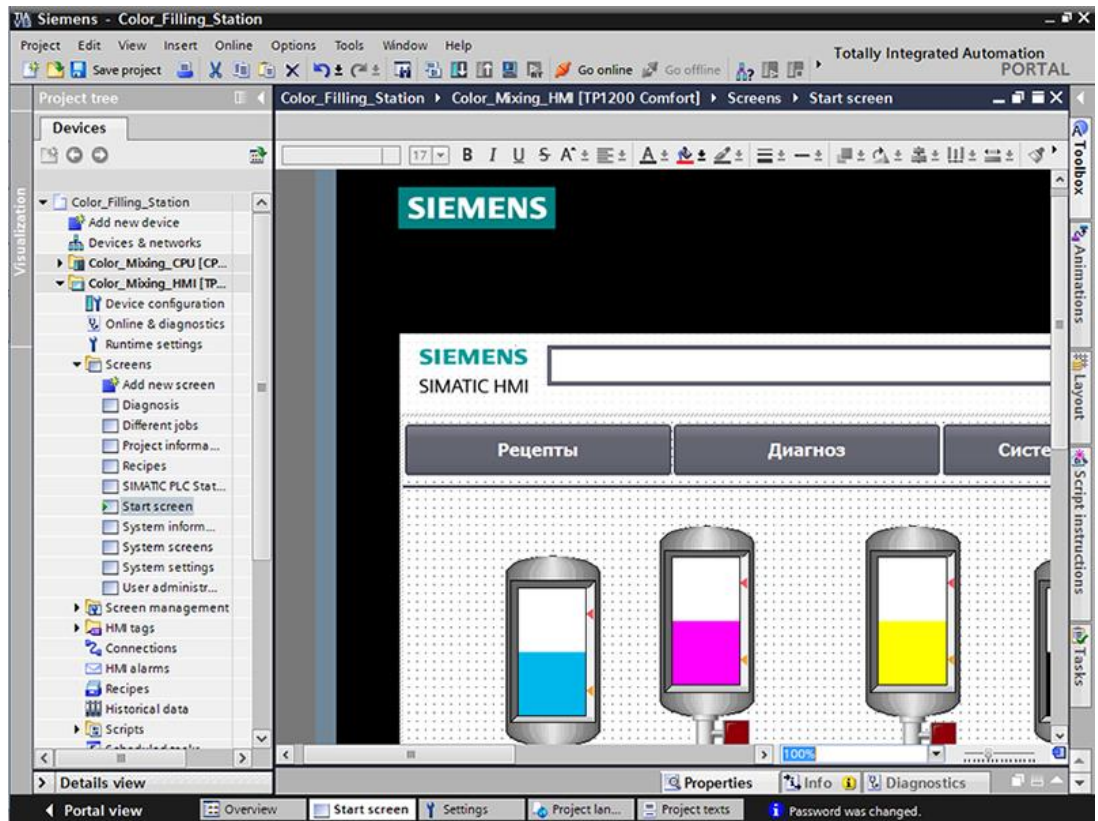
プロジェクト例は、この目的のために他の言語で表示できるように設定しています。



テキストはエクスポートされてロシア語に翻訳された後に再インポートされています。



ロシア語のテキストは言語選択の場合、ランタイム時の表示を変更することができます。

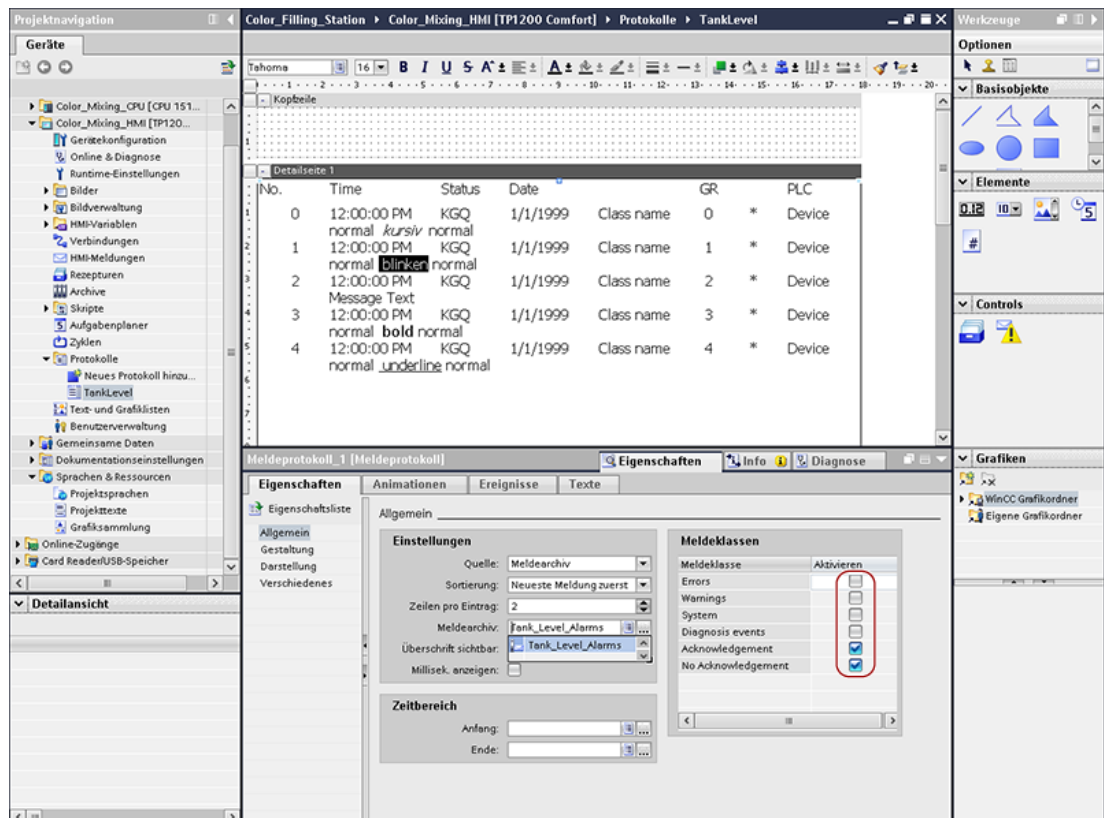


3.3.2.10 レポート

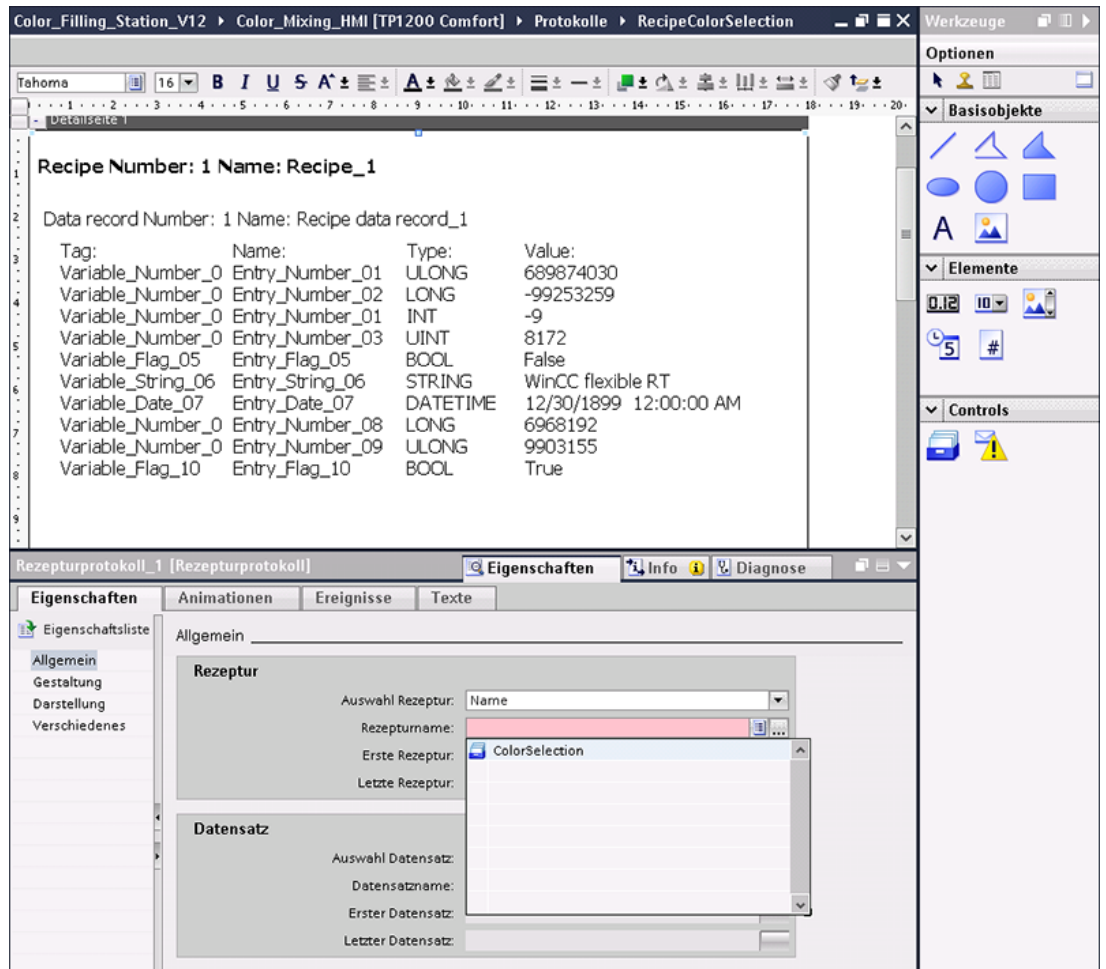
レポート

レポートは、生産試験と品質管理の基本として、生産工程でのイベントを記録するのに使用されます。アラームとレシピデータは、このためにシフトレポートの形式で定期的に出力されます。

レポートは、占有レベルのためのアラーム付きで[Tank_Level]ログ用に WinCC で作成されました。



レシピのレポートも、このプロジェクトで作成されています。



レポートは、HMI デバイスに接続されているプリンタで毎日出力することを推奨いたします。
周期的出力は、スケジューラを利用して作成可能です。

3.3.3 ライブラリからの HMI デバイスの挿入

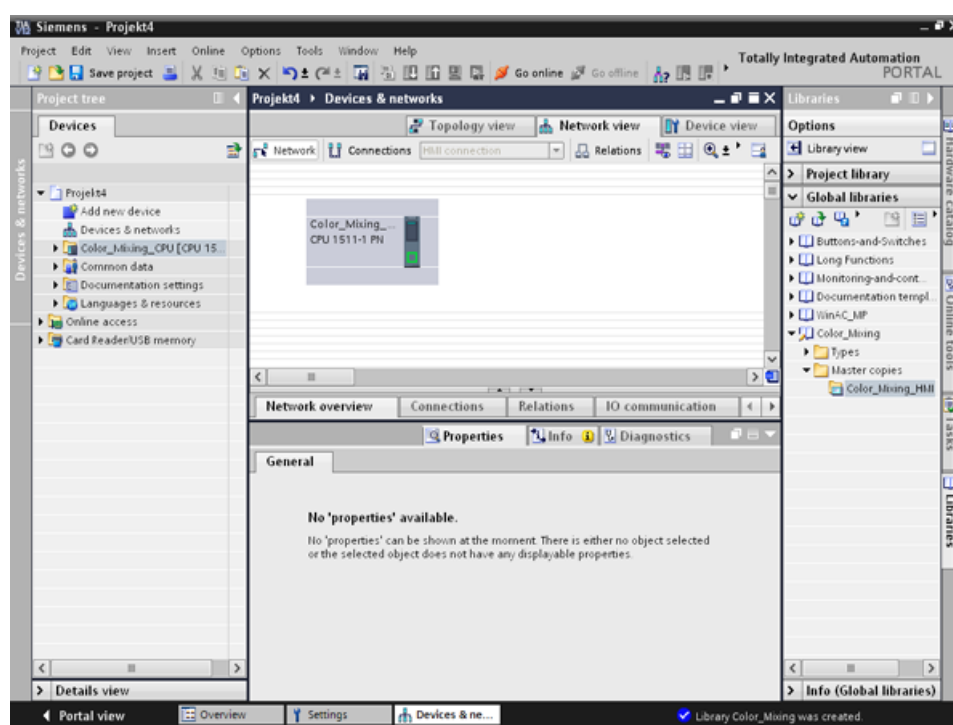
3.3.3.1 オブジェクトをライブラリに保存

概要

グローバルライブラリには、構成済み HMI デバイスが含まれます。

手順

1. グローバルライブラリを開きます。
2. HMI デバイス[Color_Mixing_HMI]を[Devices & Networks]エディタにドラッグ&ドロップします。



3. マウスポインタは、オブジェクトシンボルが付いた十字線に変わります。

結果

構成済み HMI デバイスが作成され、CPU に接続できます。

3.3.4 HMI 接続の構成定義

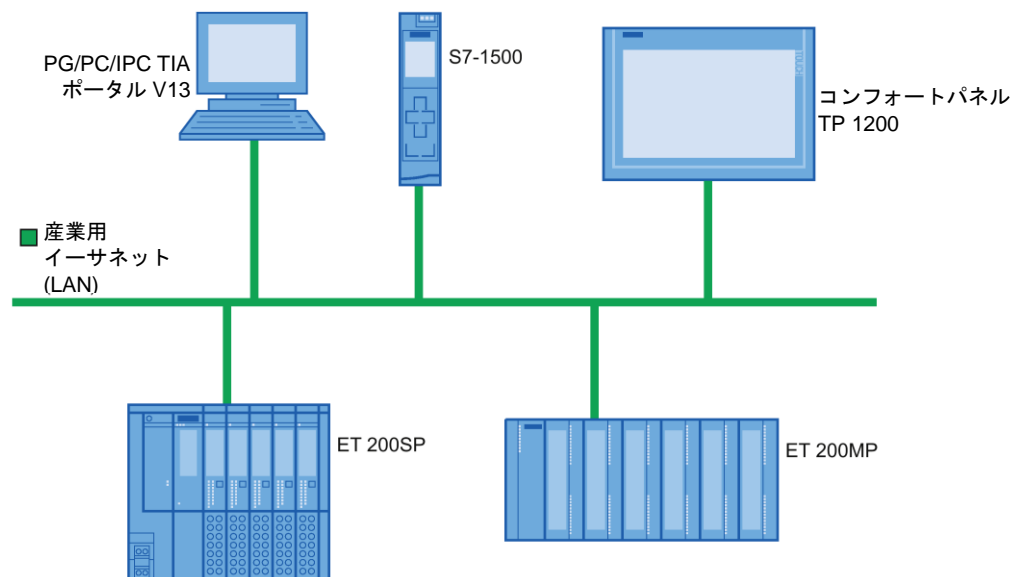
3.3.4.1 デバイス間の通信(RT Professional、RT Advanced、ベーシックパネル、パネル、コンフォートパネル、RT Unified)

通信

デバイス間のデータ交換は通信によって行われます。

デバイスは、直接、またはネットワーク経由で相互接続できます。

通信では、相互接続されたデバイスは通信パートナーと呼ばれます。



通信パートナー間のデータ転送は、多数の目的のために使用されます。

- 表示プロセス
- 操作プロセス
- アラームの出力
- プロセス値とアラームのアーカイブ
- プロセス値とアラームの文書化
- プロセスパラメータとマシンパラメータの管理

すべての通信の基本情報

すべてのタイプの通信の基本は、ネットワーク構成です。ネットワーク構成では、構成されたデバイス間に存在する接続を指定します。

また、ネットワーク構成で通信のために必要な条件を確保します。すなわち、

- ネットワークのすべてのデバイスが、固有のアドレスを割り当てられている
- デバイスは、一貫した伝送特性で通信を実行している

3.3.4.2 HMI 接続の構成定義

概要

[Devices & Networks]エディタで、コンフォートパネル TP1200 と PROFINET 経由の CPU の間の HMI 接続を構成します。

プロジェクトで使用できる通信パートナーは、ネットワークビューにおいてグラフィックで表示されます。



注意

イーサネット経由の通信

イーサネットベースの通信では、エンドユーザーがデータネットワークのセキュリティに責任を持ちます。

対象とされた攻撃はデバイスを過負荷にし、適切に機能することを妨げます。

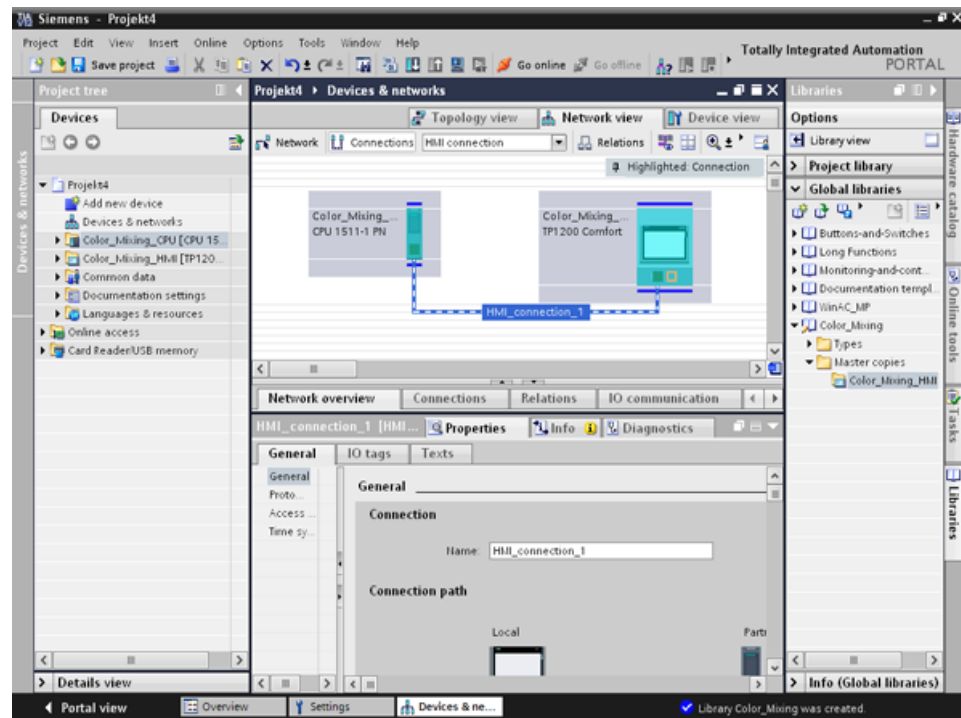
必要条件

以下の通信パートナーを[Devices & Networks]エディタで作成します。

- HMI デバイス: SIMATIC コンフォートパネル
- CPU: SIMATIC S7-1500

手順

1. [Connections]ボタンをクリックして、接続タイプとして[HMI connection]を選択します。
接続に使用できるデバイスは、色付きでハイライトされます。
2. CPU の PROFINET インターフェイスをクリックして、HMI デバイスの PROFINET インターフェイスに接続をドラッグ&ドロップします。



3. [Network view]の通信パートナーをクリックして、プロジェクトの要件に従ってインスペクタウィンドウで PROFINET パラメータを変更します。

注記

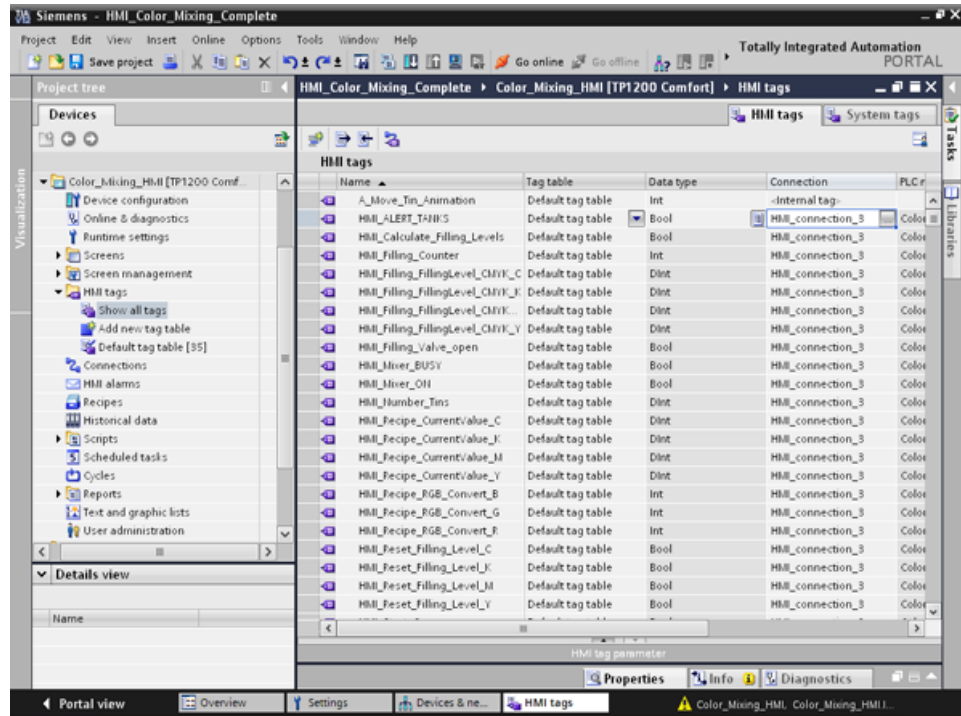
作成された HMI 接続は、また、[Connections]タブでエディタのテーブル領域に表示されます。テーブルで、接続パラメータをチェックします。

テーブルでのみ、接続のためのローカル名を変更できます。

結果

HMI デバイスと CPU の間の接続が作成されました。

3. 赤色でハイライトされているすべてのエントリに対してこの手順を繰り返します。



結果

HMI 接続が、CPU と HMI デバイスで既に構成されたタグ用に作成されました。
この HMI 接続が確立しました。

3.3.5 システム診断の構成定義

3.3.5.1 システム診断の基本(RT Advanced、パネル、コンフォートパネル)

概要

システム診断を使用して、プラントのあらゆる部分の問題やエラーを検出します。WinCC には、素早いエラーの特定のために 2 つの表示エレメントとオペレーティングエレメントがあります。

システム診断ビュー

アラームビューは CPU のステータスを表示し、一方システム診断ビューはシステムで利用できるすべてのデバイスの概要を表示します。これで、エラーの原因と関連デバイスが判明します。[Devices & networks]エディタで構成した、診断をサポートするすべてのデバイスへアクセスできます。

システム診断ウィンドウ

システム診断ウィンドウは操作および表示エレメントで、グローバル画面でだけ使用できます。

システム診断ウィンドウのファンクションは、システム診断ビューのファンクションと違いはありません。システム診断ウィンドウがグローバル画面で構成されるため、オブジェクトがランタイム時にクローズ可能な場合、指定できます。

概要

- デバイスビュー
- 診断バッファビュー
- 詳細ビュー
- マトリックスビュー(マスタシステム、PROFIBUS、PROFINET のみ)

デバイスビューは、レイヤで利用できるすべてのデバイスを表で表示します。デバイスをダブルクリックして、子デバイスまたは詳細ビューを開きます。最初の列のシンボルは、デバイスの現在のステータスについての情報を表示します。

[illegible]

診断バッファビュー

診断バッファからの現在のデータが、診断バッファビューに表示されます。

S7-1500-Station_1 \ CPU-Proxy_1 \ Diagnostic buffer view

N.	Date	Time	Event	
1	1/11/2012	10:49:59 PM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...	✓
2	1/11/2012	10:49:59 PM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...	✓
3	1/11/2012	10:49:56 PM	Communication initiated request: STOP - CPU changes from RUN to STOP mode	✓
4	1/11/2012	7:32:36 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...	✓
5	1/11/2012	7:32:36 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...	✓
6	1/11/2012	7:32:34 AM	Communication initiated request: STOP - CPU changes from RUN to STOP mode	✓
7	1/11/2012	7:30:53 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...	✓
8	1/11/2012	7:30:53 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...	✓
9	1/11/2012	7:21:03 AM	Communication initiated request: STOP - CPU changes from RUN to STOP mode	✓
...	1/11/2012	4:33:55 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...	✓
...	1/11/2012	4:33:55 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...	✓
...	1/11/2012	4:22:31 AM	Follow-on operating mode change - CPU changes from STOP (initialization) to STOP mode	✓
...	1/11/2012	4:22:31 AM	Power on - CPU changes from NO POWER to STOP (initialization) mode	✓
...	1/11/2012	4:21:59 AM	Power off - CPU changes from STOP to NO POWER mode	✓
...	1/11/2012	4:19:51 AM	System initiated request: STOP - CPU changes from STOP (firmware update) to STOP mode	✓
...	1/11/2012	4:19:51 AM	Device firmware update finished - CPU changes to STOP mode, new startup inhibit set: .	✓
...	1/11/2012	4:19:51 AM	CPU firmware update: Completed successfully - Device firmware update sequence contin...	✓
...	1/11/2012	4:19:03 AM	System initiated request: Firmware update - CPU changes from STOP (initialization) to ST...	✓

詳細ビュー

詳細ビューは、選択したデバイスとすべての保留中のエラーについての詳細情報を表示します。詳細ビューで、データが正しいか確認します。詳細ビューではエラーテキストをソートできません。

S7-1500 station_1 \ DI32

› Status	
› Name	DI32
› Betriebszustand	
› Baugruppenträger	0
› Steckplatz	3
› Typ	DI32
› Bestellnummer	6ES7 521-1BL00-0AB0
› Adresse	259*
› Anlagenbezeichnung	
› Positionskenung	
› Hersteller ID	SIEMENS AG
› Hardware Version	97
› Profil ID	
› spezifische Profildaten	0003
› I&M Datenversion	1.1
› Fehlertext	Drahtbruch
	Hilfe: Kontrollieren Sie den Zustand der Verbindungsleitungen.






マトリックスビュー

マトリックスビューは、マスタシステムでだけ利用できます。マトリックスビューは、マスタシステムのサブデバイスのステータスを表示します。

- PROFIBUS では、PROFIBUS で割り当てた番号を識別番号(DP ステーション番号)として使用します。
- IO デバイスは、PROFINET で 1 から連続的に番号付けされています。



ナビゲーションボタン

ボタン	ファンクション
	子デバイスがない場合、子デバイスまたは詳細ビューを開きます。
	親デバイスがない場合、親デバイスまたはデバイスビューを開きます。
	デバイスビューを開きます。
	診断バッファビューを開きます。デバイスビューでのみ表示できます。
	ビューを更新します。 F2 などの構成されたソフトキー

3.3.5.3 システム診断ビューの構成定義(RT Advanced、ベーシックパネル、パネル、コンフォートパネル)

概要

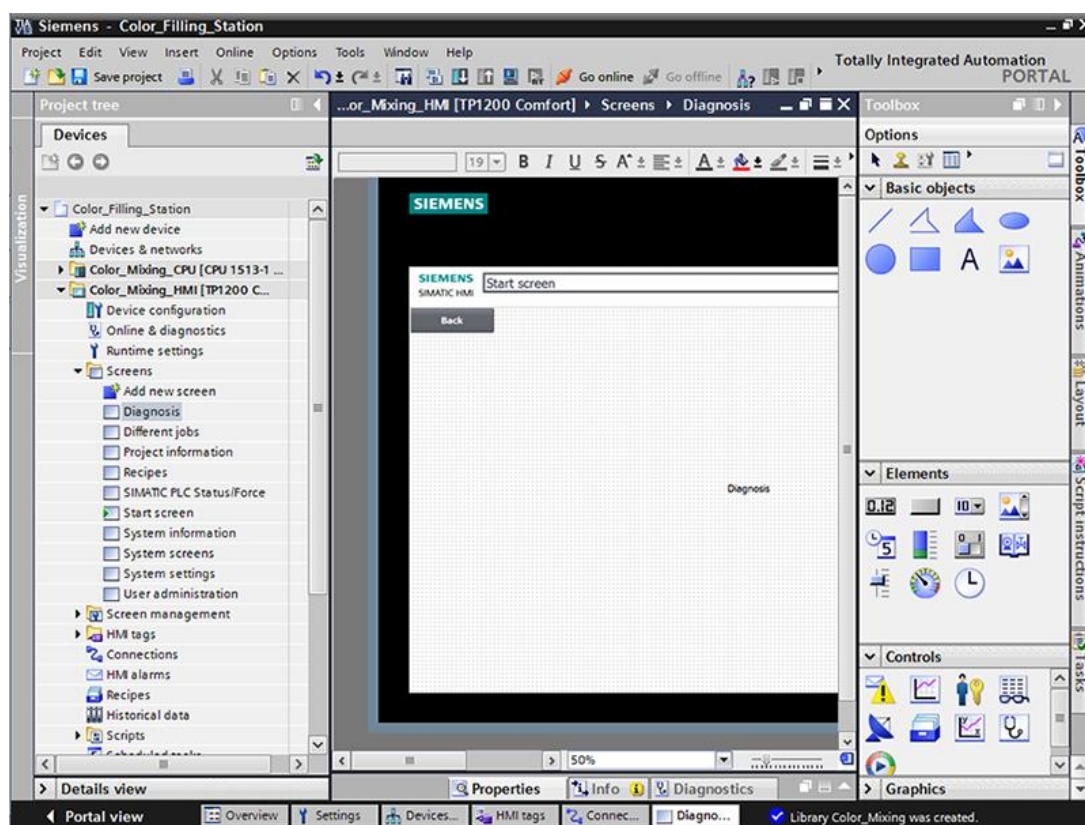
システム診断ビューをプロジェクトに追加して、プラント内で使用可能なすべてのデバイスの概要を表示します。

必要条件

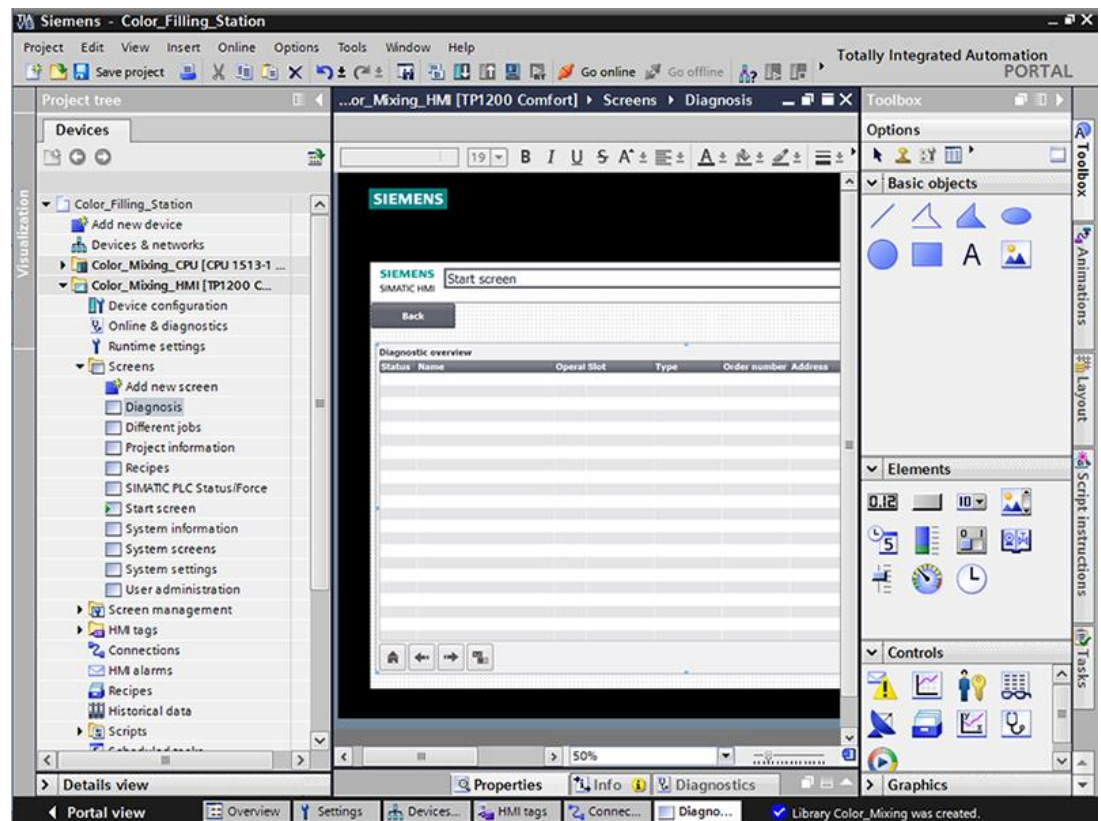
- CPU が作成されていること。
- インспекタウィンドウが開いていること。

手順

1. [Diagnostics]HMI 画面をダブルクリックします。



2. [ツール]タスクカードで、[システム診断ビュー]オブジェクトをダブルクリックします。
画面にオブジェクトが追加されます。



3. インспекタウィンドウで、[Properties] → [Properties] → [Columns] → [Devices/Detail view]を選択します。
4. [State]、[Name]、[Slot]などのランタイム時にデバイスビューで必要となる列を有効にします。
5. ランタイム時に詳細ビューで必要となる列([State]、[Name]、[Slot]、[Higher level designation]など)を有効にします。
6. 診断バッファビューで必要な列([State]、[Name]、[Rack]など)を有効にします。
7. 必要に応じて、ヘッダーを列に合わせて調整します。
8. [Properties] → [Properties] → [Layout] → [Column settings] → [Columns moveable]を有効にし、ランタイム時に列を移動します。
9. 必要に応じて、[Properties] → [Properties] → [Column headers]で列ヘッダーを変更できます。

結果

システム診断ビューが[Diagnostics]画面に追加されました。

これで、プラント全体のエラーメッセージが、ランタイム時にシステム診断ビューに表示されます。

3.3.6 HMI デバイスのシミュレーション

3.3.6.1 シミュレーションの基本(RT Professional、RT Advanced、パネル、コンフォートパネル)

概要

シミュレータを使用して、設定用 PC でコンフィグレーションのパフォーマンスを試験できます。これにより、運用操作前に論理的な構成エラーをすぐに特定できます。

以下のようにしてシミュレータを開始できます。

- HMI デバイスのショートカットメニューまたは画面、[Start simulation]で
- メニューコマンド[Online] → [Simulation] → [Start|With tag simulator|With script debugger]で
- ポータルビューの[Visualization] → [Simulate device]で

必要条件

シミュレーション/ランタイムコンポーネントが、設定用 PC にインストールされていること。

応用分野

シミュレータを使用して、HMI システムの以下のようなファンクションを試験できます。

- 制限レベルおよびアラーム出力のチェック
- 割り込みの一貫性
- 構成された割り込みシミュレーション
- 構成された警告
- 構成されたエラーメッセージ
- ステータス表示のチェック

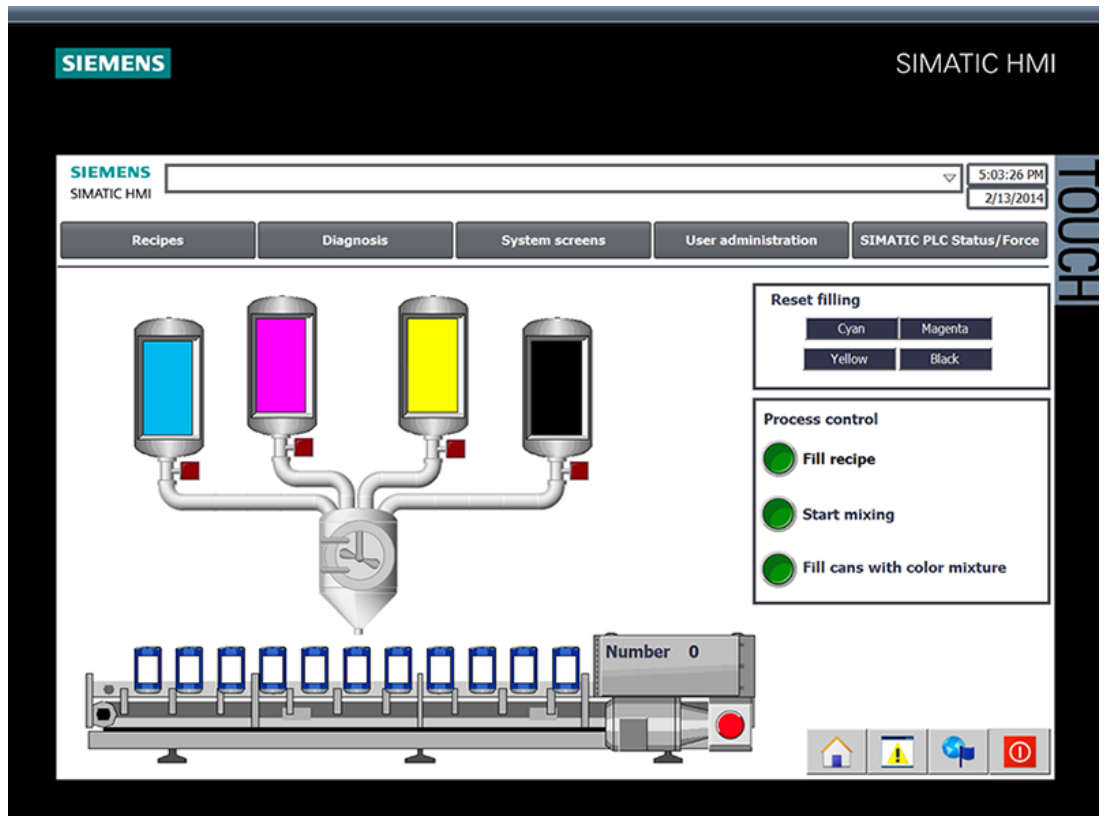
3.3.6.2 シミュレーションでパネルの操作

概要

ご使用のコンピュータで HMI プロジェクトをシミュレートします。

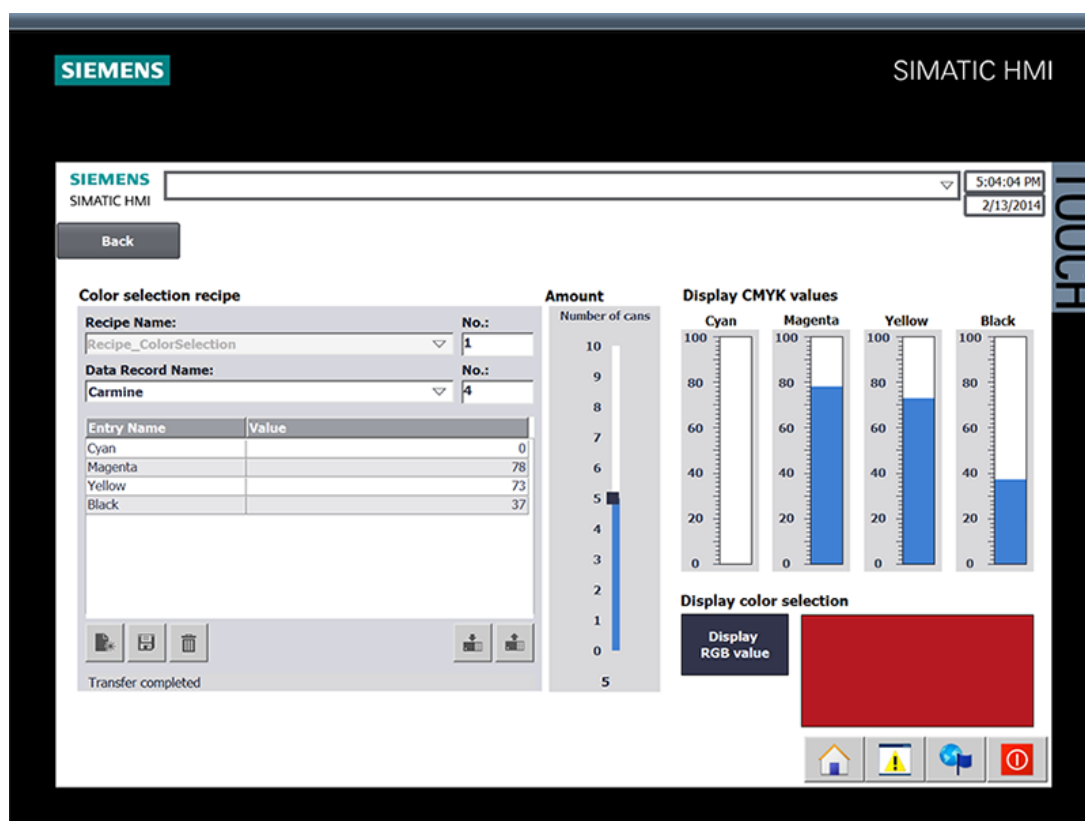
手順

1. HMI デバイスのシミュレーションを開始します。

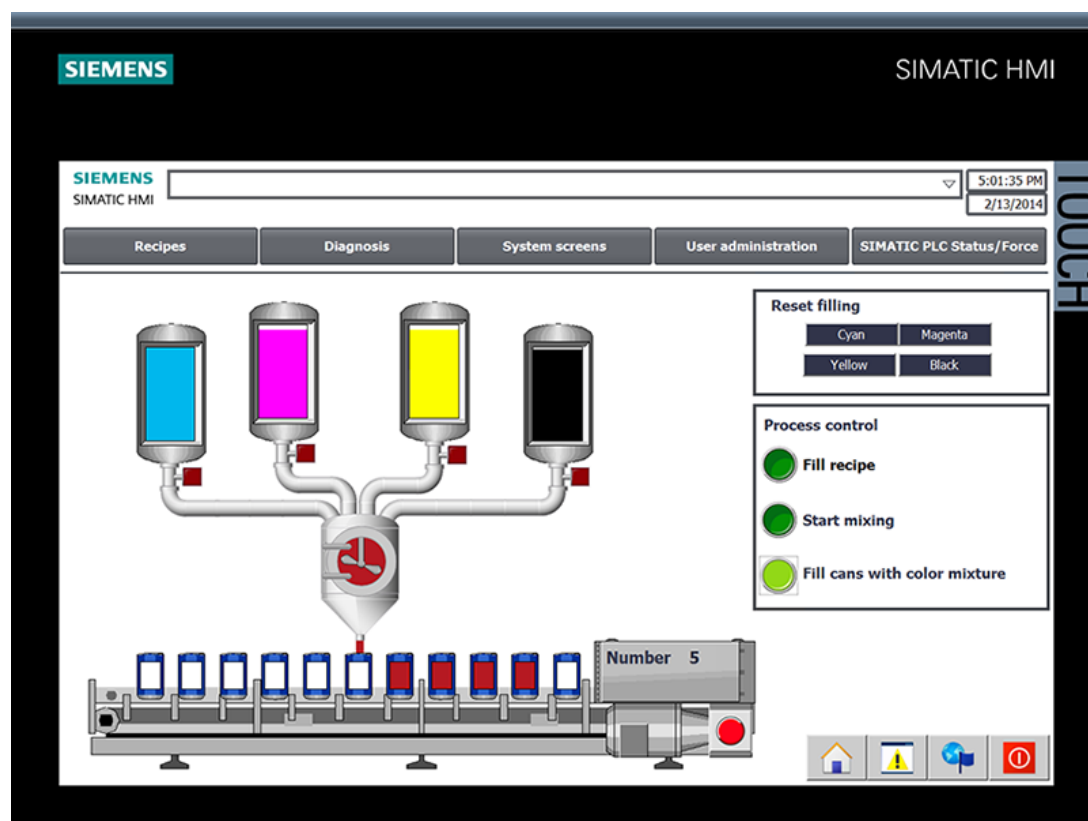


- CPU の接続が確立され、カラーミキシングシステムがシミュレーションで表示されます。
2. [Recipes]画面を開き、色を選択します。

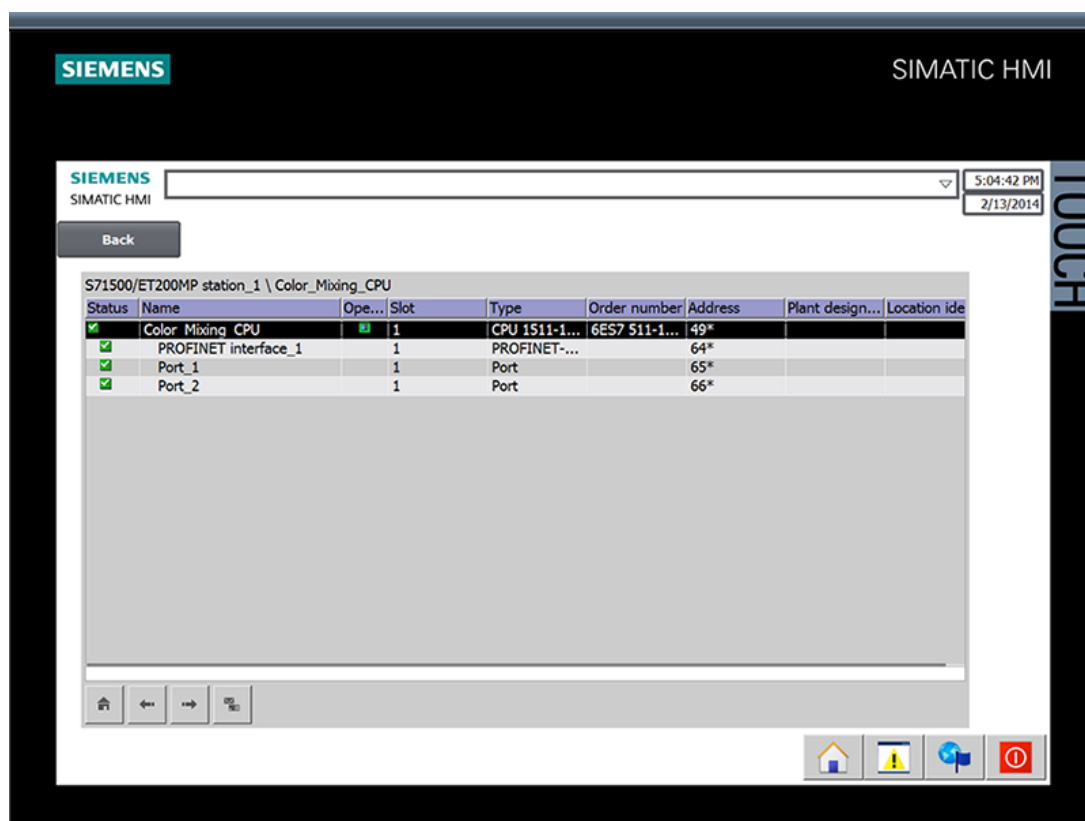
3. 缶数を指定し、選択した色を表示します。



4. 開始画面に戻り、生産を開始します。



5. [Diagnostics]画面で現在の CPU ステータスを照会できます。



3.4 プロジェクトをプログラミングツールにアップロード

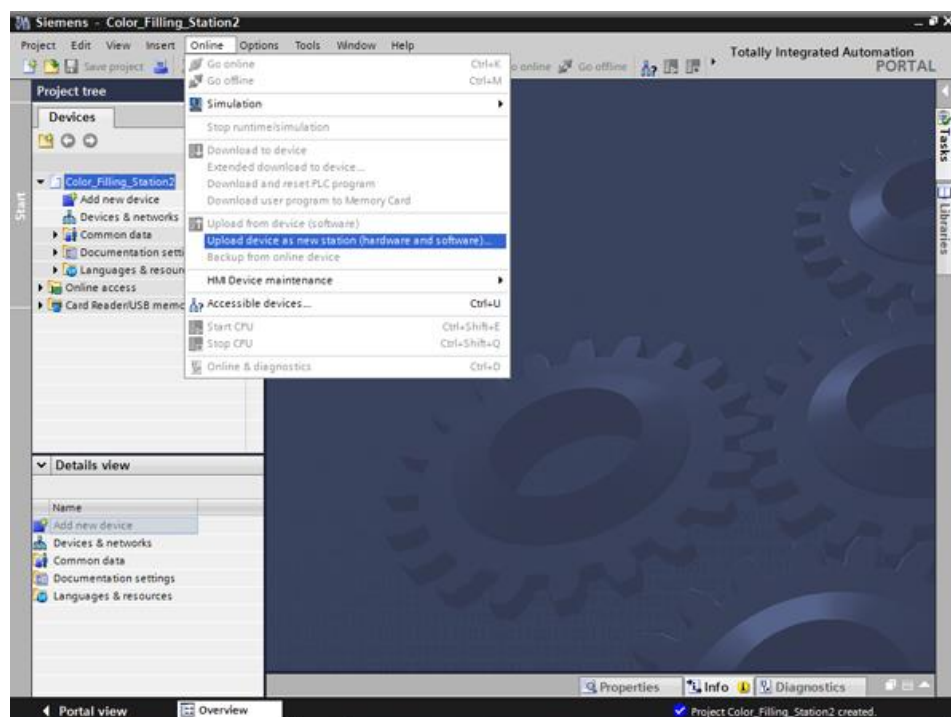
3.4.1 CPU をプロジェクトにアップロード

概要

ハードウェアコンフィグレーションとユーザープログラムからの現在値を含む新しいステーションを作成できます。

手順

1. CPU からアップロードするためにダイアログを開きます。

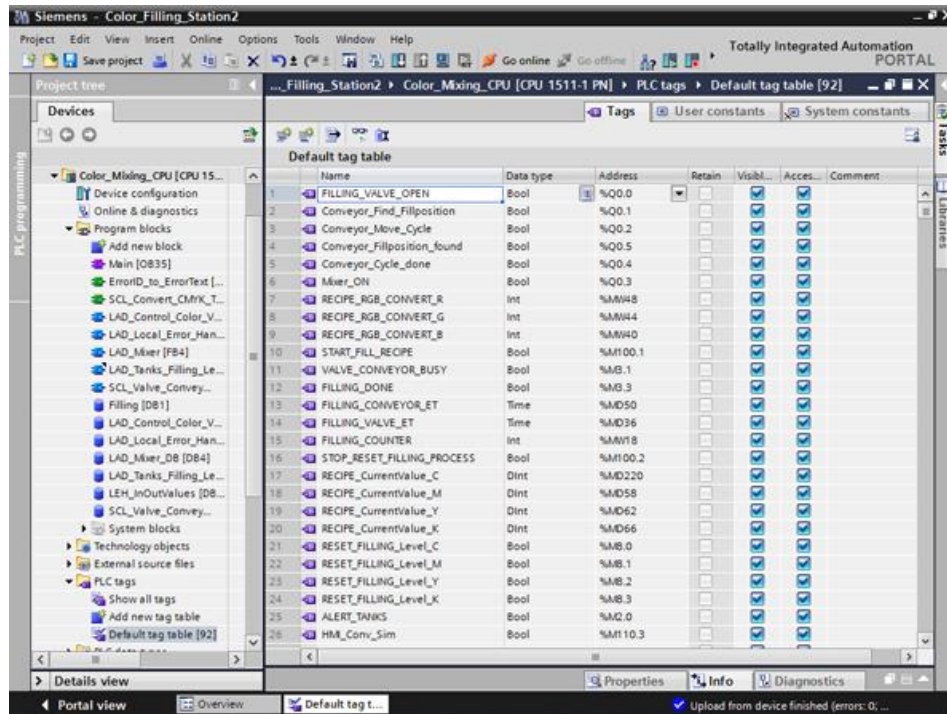


2. プログラミングツールを CPU に接続するインターフェースを選択します。アクセス可能なノードの検索が自動的に始まります。
3. CPU をプロジェクトにアップロードします。

3.4 プロジェクトをプログラミングツールにアップロード

結果

CPU のハードウェア構成とソフトウェア設定をプロジェクトにアップロードします。
これで、プロジェクトにはプログラムブロックやタグなどが含まれます。



3.5 Inter Project Engineering による Team Engineering

3.5.1 「Inter Project Engineering」の基本

概要

このセクションでは、Team Engineering の利点と HMI プロジェクトエンジニア用に必要な CPU データの作成方法について学びます。HMI プロジェクトエンジニアとして、プロジェクトでこの CPU データをどのように使うかも学びます。

リモート構成

[Inter Project Engineering]を使用して、別々の場所でユーザープログラムとユーザーインターフェースを平行して開発できます。HMI プロジェクトエンジニアは、CPU ユーザープログラムを必要としません。STEP 7 インストールは不要です。

CPU インターフェースのタグ、ブロック、メッセージおよびアドレス情報だけが、HMI デバイスと CPU 接続のために究極的に関係します。都合のよいことに、プログラマはこのデータを IPE ファイルにエクスポートすることができ、ファイルは HMI 開発者によってプロジェクトにインポートされます。更新は、新しい IPE ファイルを転送することによっていつでも可能です。

CPU と HMI デバイスにダウンロード後も、データは一貫しています。HMI 構成で作成された CPU との接続は、最新のままです。

注記

V5.4 SP3 移行から、TIA ポータルにインポートして、新しい第 2 世代のベーシックパネルとコンフォートパネルを STEP 7 プロジェクトに統合できます。

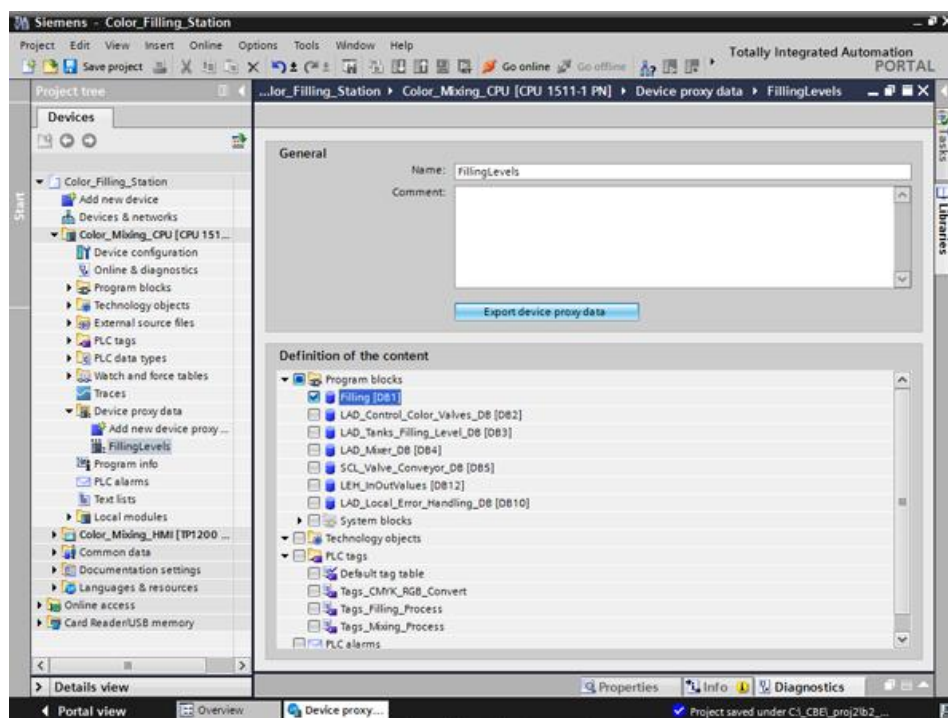
3.5.2 IPE ファイルの作成

概要

コンパクトな HMI デバイスを使用して、占有レベルを直接、塗料混合プラントで表示することができます。ビジュアライゼーションのためにエンジニアリング会社と契約し、必要な CPU データを IPE ファイルとして提供します。

手順

1. CPU に新しいプロキシデータを追加します。
2. 名前を入力し、必要な CPU データを選択します。



3. プロキシデータをエクスポートします。

結果

IPE ファイルが作成されます。電子メールなどで IPE ファイルをエンジニアリング会社へ送信できます。

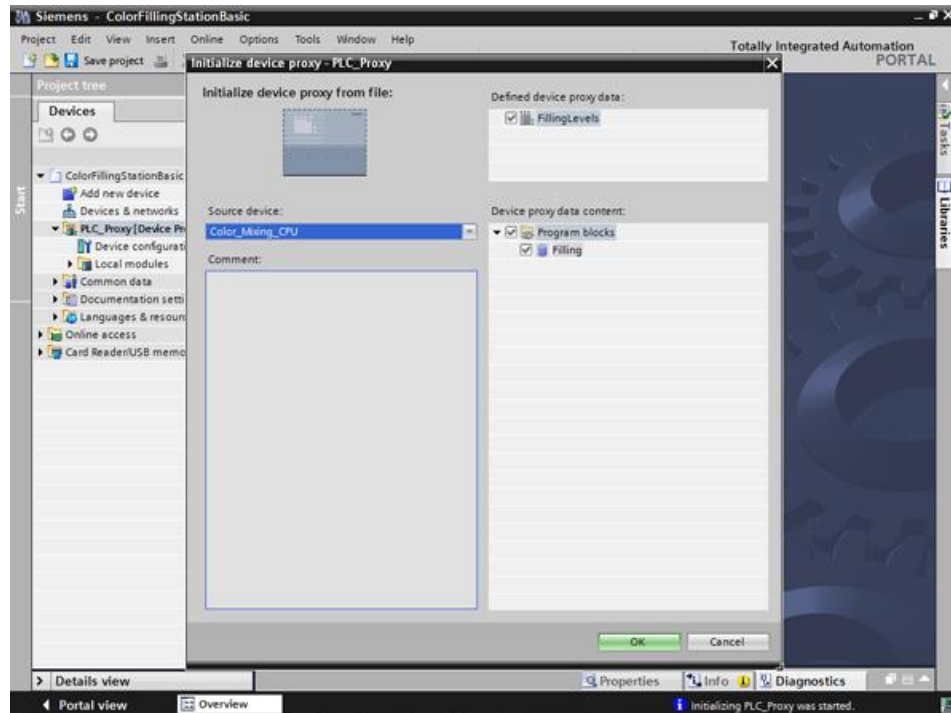
3.5.3 IPE ファイルのインポート

概要

エンジニアリングオフィスでは、プロジェクトエンジニアが新しいプロジェクトにデバイスプロキシを作成し、IPE ファイルからの CPU データで初期化します。プロジェクトエンジニアは、IPE ファイルの更新ごとに初期化を繰り返します。

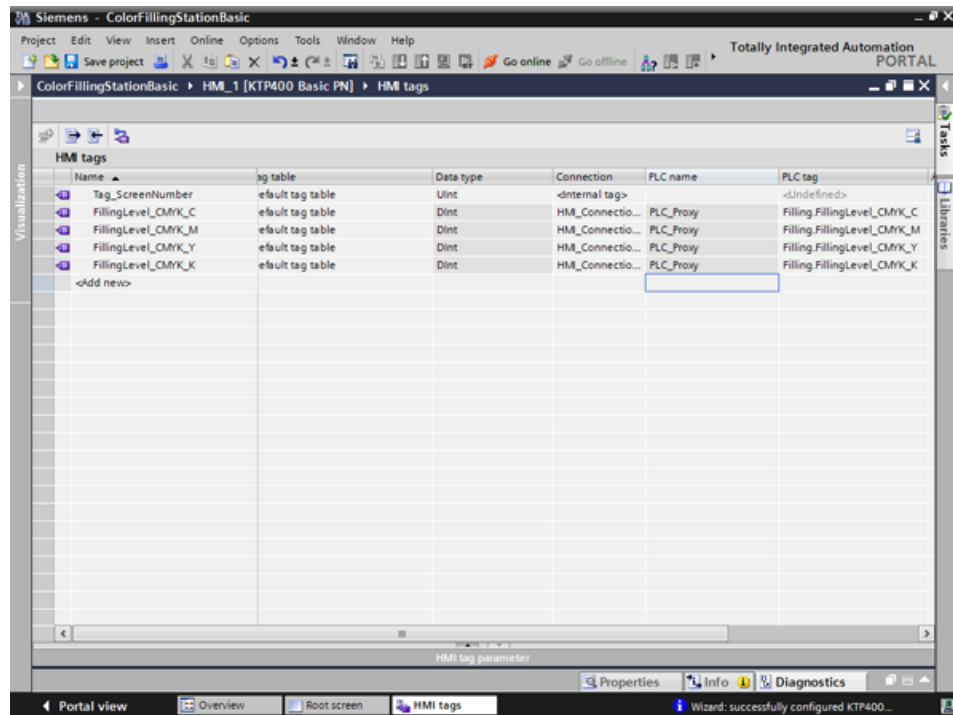
手順

1. 新しいプロジェクトに CPU 用のデバイスプロキシを作成し、それを初期化します。



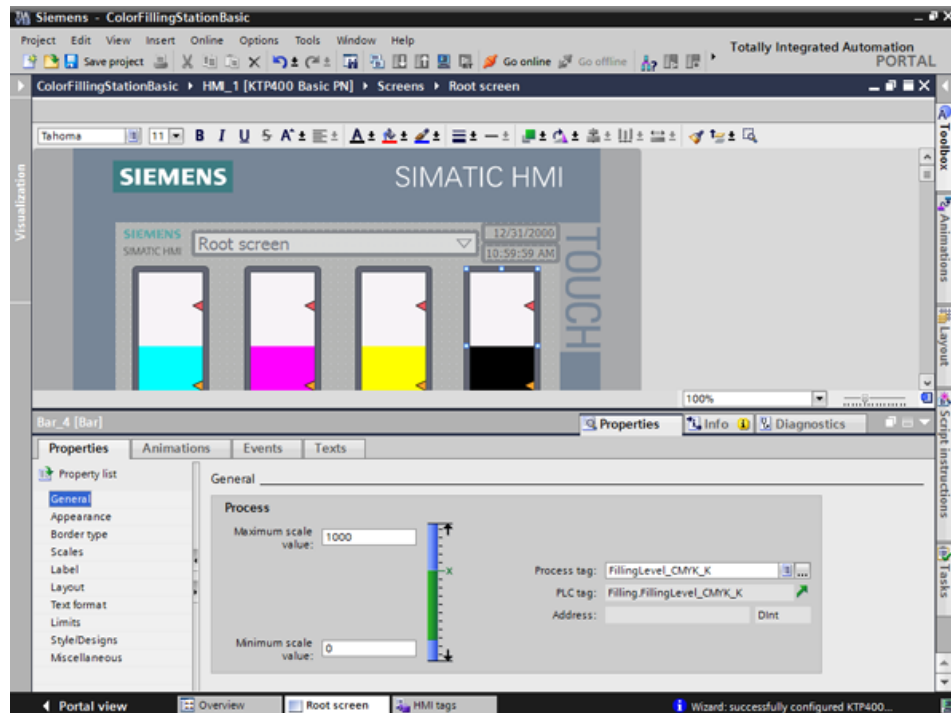
2. デバイスウィザードを使用して、ベーシックパネルを挿入します。
3. [Cyan]色の占有レベルのための HMI タグを作成し、PLC タグを選択します。

4. 他の HMI タグも同じ方法で作成します。



5. 棒グラフを作成して、[Cyan]色の占有レベルを表示します。

6. 他の占有レベル用の棒グラフも同じ方法で作成します。



7. プロジェクトをコンパイルします。

結果

これで、プロジェクトは、コミッショニングエンジニアによって HMI デバイスへダウンロードできます。CPU との通信は、IPE ファイルからの CPU データのおかげで最新の状態です。

セキュリティ

4.1 CPU の保護ファンクションの概要

概要

この章では、未承認のアクセスから S7-1500 オートメーションシステムを保護するための以下のファンクションについて説明します。

- アクセス保護
- ノウハウプロテクト
- コピー保護
- CPU ロックによる保護

その他の CPU 保護対策

以下の対策により、S7-1500 CPU のファンクションおよびデータへのネットワーク経由の外部ソースからの未許可のアクセスに対する保護がさらに強化されます。

- Web サーバーの無効化
- NTP サーバー経由の時刻同期の無効化
- PUT/GET 通信の無効化

Web サーバーを使用する場合、ユーザー管理で特定のユーザーに、パスワードで保護されたアクセス権を設定することによって、未承認のアクセスに対して S7-1500 オートメーションシステムを保護します。

4.2 ディスプレイを使用して、追加のアクセス保護を設定

概要

S7-1500 の表示で、パスワードで保護された CPU へのアクセスをブロックすることができます(ローカルブロック)。アクセスロックが有効なのは、動作モードスイッチが RUN 位置の場合だけです。アクセスロックはパスワード保護とは独立して適用されます。すなわち、何者かが接続されたプログラミングツール経由で CPU にアクセスし、正しいパスワードを入力し、CPU にアクセスしたとしても、まだブロックされます。たとえば、読み取りアクセスはローカルに許可されるが、書き込みアクセスはローカルに許可されないように、ディスプレイで各アクセスレベルに対して、アクセスブロックを別々に設定することができます。

手順

パスワードによるアクセスレベルが STEP 7 で設定されている場合、ディスプレイを使用してアクセスをブロックすることができます。

ディスプレイで S7-1500 CPU のローカルアクセス保護を設定するには、以下のように実行します。

1. ディスプレイで、[設定|保護]メニューを選択します。
2. [OK]で選択を確定し、各アクセスレベルに対して、RUN モードセクタでアクセスが許可されるかどうかを設定します。

許可: STEP 7 で一致するパスワードが入力される場合、CPU へのアクセスは可能です。

RUN モード時に無効化: 動作モードスイッチが RUN 位置の場合、パスワードを知っていたとしても、このアクセスレベルの権限を持つユーザーは CPU にログインできません。STOP モード時には、アクセスはパスワード入力によって可能になります。

表示のアクセス保護

ローカルアクセス保護がローカルパスワードによって保護されるように、CPU のプロパティで STEP 7 での表示のためのパスワードを設定することができます。

4.3 ノウハウプロテクト

ノウハウプロテクトを使用して、未承認のアクセスから、プログラム内の OB、FB、FC タイプの 1 つ以上のブロックとグローバルデータブロックを保護することができます。ブロックへのアクセスを制限するためにパスワードを入力できます。パスワード保護は、ブロックが承認なしで読み取られたり、変更されたりするのを防ぎます。

パスワードがない場合、ブロックに関する以下のデータだけを読み出すことができます。

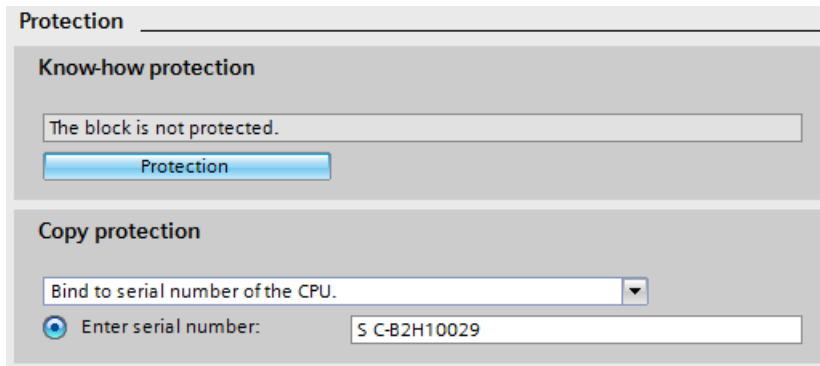
- ブロックタイトル、ブロックコメント、およびブロックプロパティ
- ブロックパラメータ (INPUT、OUTPUT、IN、OUT、RETURN)
- プログラムの呼び出し構造体
- 使用場所に関する情報のないグローバルタグ

ノウハウプロテクトされたブロックの場合、さらに以下の操作を実行できます。

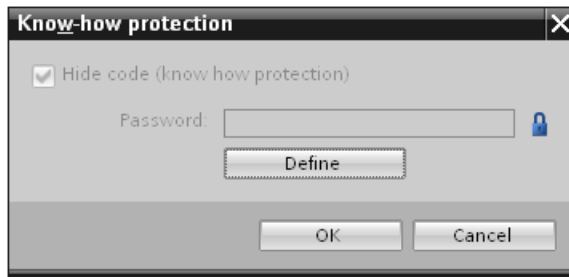
- コピーと削除
- プログラムの呼び出し
- オンライン/オフライン比較
- 読み込み

ブロックのノウハウプロテクトの設定

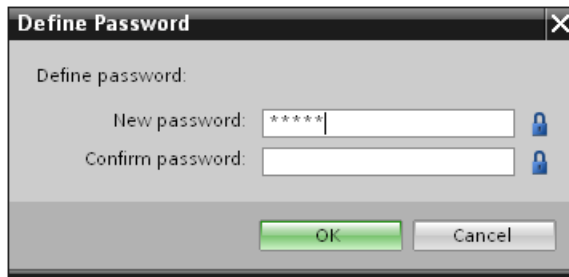
1. 個々のブロックのプロパティを開きます。
2. [General]で[Protection]オプションを選択します。



3. [Protection]ボタンをクリックして、[Know-how protection]ダイアログを表示します。



4. [Define]ボタンをクリックして、[Define Password]ダイアログを開きます。



5. [New password]フィールドに新規パスワードを入力します。[Confirm password]フィールドに同じパスワードを入力します。
6. [OK]をクリックして入力を確定します。
7. [OK]をクリックして[Know-how protection]ダイアログを閉じます。

結果: 選択されたブロックがノウハウプロテクトされます。ノウハウプロテクトされたブロックは、プロジェクトツリーで錠がついています。入力されたパスワードは選択されたすべてのブロックに適用されます。

ノウハウプロテクトされたブロックを開く

1. ブロックをダブルクリックして、[Access protection]ダイアログを開きます。
2. ノウハウプロテクトされたブロックのパスワードを入力します。
3. [OK]をクリックして入力を確定します。

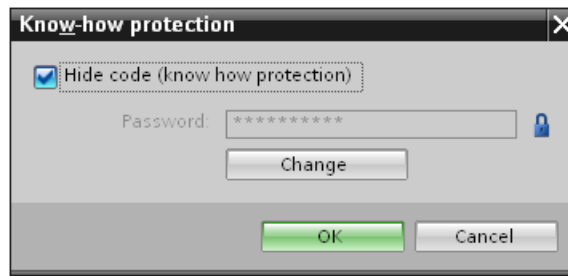
結果: ノウハウプロテクトブロックが開きます。

ブロックを開いたら、ブロックまたは TIA ポータルが開いている限り、ブロックのプログラムコードとブロックインターフェースを編集できます。次にブロックを開くときは、再度パスワードを入力してください。[Access protection]ダイアログを[Cancel]で閉じた場合、ブロックは開きますがブロックコードは表示されず、ブロックを編集することはできません。

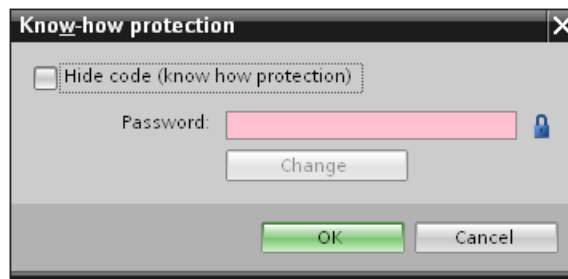
ブロックのコピーやライブラリの追加などの場合でも、ブロックのノウハウプロテクトは解除されません。コピーもノウハウプロテクトされます。

ブロックのノウハウプロテクトの削除

1. ノウハウプロテクトを削除するブロックを選択します。保護されたブロックは、プログラムエディタでは開けません。
2. [Edit]メニューで、[Know-how protection]コマンドを選択して[Know-how protection]ダイアログを開きます。
3. [Hide code (Know-how protection)]チェックボックスを無効にします。



4. パスワードを入力します。



5. [OK]をクリックして入力を確定します。

結果: 選択したブロックからノウハウプロテクトが削除されます。

4.4 コピー保護の設定

コピー保護で、プログラムやブロックを特定の SIMATIC メモリカードや CPU に結びつけることができます。SIMATIC メモリカードまたは CPU のシリアル番号のリンクにより、このプログラムまたはこのブロックは特定の SIMATIC メモリカードまたは CPU との組み合わせでのみ使用可能となります。このファンクションにより、プログラムやブロックを電氣的に送信可能(電子メールなど)、またはメモリモジュールによって出荷できます。

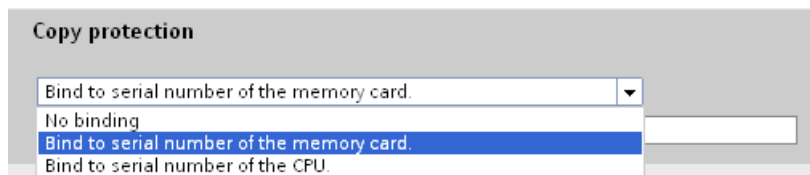
ブロックにそのようなコピー保護をセットアップする場合、ノウハウプロテクトをこのブロックにも割り当てます。ノウハウプロテクトがない場合、だれもがコピー保護をリセットできます。ただし、ブロックが既にノウハウプロテクトされている場合、コピー保護設定を読み取り専用として最新にコピー保護をセットアップする必要があります。

コピー保護の設定

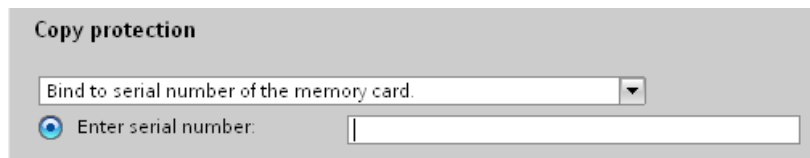
1. 個々のブロックのプロパティを開きます。
2. [General]で[Protection]オプションを選択します。



3. [Copy protection]領域で、ドロップダウンリストから[Bind to serial number of the CPU]または[Bind to serial number of the memory card]を選択します。



4. CPU または、SIMATIC メモリカードのシリアル番号を入力します。



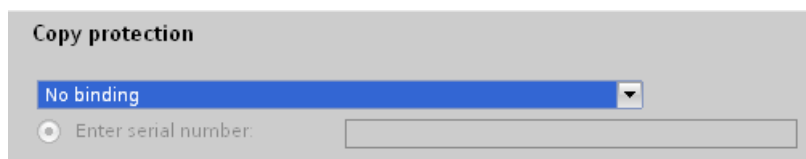
5. [Know-how protection]領域でブロックのノウハウプロテクトを設定できるようになりました。

注記

指定されたシリアル番号に一致しないデバイスにノウハウプロテクトされたブロックのコピーをダウンロードする場合、ダウンロード操作全体が拒否されます。コピー保護されていないブロックもダウンロードされません。

コピー保護の削除

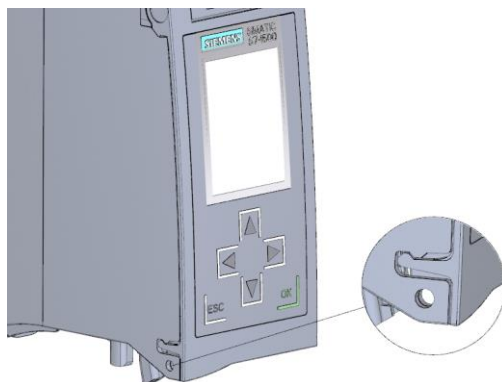
1. 既存のノウハウプロテクトをすべて削除します。
2. 個々のブロックのプロパティを開きます。
3. [General]で[Protection]オプションを選択します。
4. [Copy protection]領域で、ドロップダウンリストから[No binding]を選択します。



4.5 CPU ロックによる保護

十分にセキュアなフロントカバーを使用して、未承認のアクセスから CPU を保護します。
CPU カバーのラッチを使用した以下のオプションがあります。

- シールの貼り付け
- フロントカバーをロックで固定(シャックル直径: 3 mm)



4.6 CPU のアクセス保護の設定

概要

CPU は特定のファンクションへのアクセスを制限するために 4 つのアクセスレベルを提供します。

CPU のアクセスレベルとパスワードの設定によって、パスワードを入力せずにアクセスできるファンクションとメモリ領域を制限できます。個々のアクセスレベルならびに関連パスワードの入力は、CPU のオブジェクトプロパティで指定します。

CPU のアクセスレベル

アクセスレベル	アクセス制限
Full access (No protection)	ハードウェアコンフィグレーションとブロックは、すべてのユーザーによって、読み取り、変更することができます。
Read access	このアクセスレベルを使用すると、ハードウェアコンフィグレーションおよびブロックへの読み取り専用アクセスが、パスワードを入力せずに可能です。つまり、ハードウェアコンフィグレーションおよびブロックをプログラミングツールにアップロードすることができます。HMI アクセスおよび診断データへのアクセスも可能です。 パスワードを入力せずに、ブロックまたはハードウェアコンフィグレーションを CPU にダウンロードすることはできません。また、パスワードなしで以下を行うことはできません: テストファンクションの書き込み、動作モード(RUN/STOP)の変更、およびファームウェアの更新(オンライン)。
HMI Access	このアクセスレベルを使用すると、診断データへの HMI アクセスおよびアクセスだけは、パスワードを入力せずに使用できます。 パスワードを入力せずに、ブロックおよびハードウェアコンフィグレーションを CPU にダウンロードすることも、ブロックおよびハードウェアコンフィグレーションを CPU からプログラミングツールにアップロードすることもできません。また、パスワードなしで以下を行うことはできません: テストファンクションの書き込み、動作モード(RUN/STOP)の変更、およびファームウェアの更新(オンライン)。
No access (complete protection)	CPU が完全に保護されている場合、ハードウェアコンフィグレーションとブロックへの読み取りアクセスまたは書き込みアクセスは使用できません。HMI アクセスも不可能です。PUT/GET 通信のサーバーファンクションは、このアクセスレベルでは無効です(変更できません)。 パスワードによる承認によって、CPU へのフルアクセスが再度可能になります。

各アクセスレベルによって、パスワードを入力せずに特定のファンクション([Accessible devices]ファンクションを使用する識別など)に制限なしでアクセスすることができます。

CPU の初期設定は[制限なし]と[パスワード保護なし]です。CPU へのアクセスを保護するために、CPU のプロパティを編集してパスワードを設定する必要があります。

CPU 間の通信(ブロックの通信ファンクションを使用)は、PUT/GET 通信が無効になっていない限り、CPU の保護レベルに制限されません。

正しいパスワードの入力により、対応するレベルで許可されているすべてのファンクションにアクセスすることができます。

注記

アクセスレベルの構成定義はノウハウプロテクトを置換しません

アクセスレベルを設定することは、ダウンロード権限を制限することによって CPU への許可されない変更を防ぎます。ただし、SIMATIC メモリカードのブロックは、書き込み保護または読み取り保護されません。ノウハウプロテクトを使用して、SIMATIC メモリカードのブロックのコードを保護します。

アクセスレベルの手順のパラメータ設定

S7-1500 CPU のアクセスレベルを設定するには、次のように実行します。

1. インспекタウィンドウで、S7-1500 CPU のプロパティを開きます。
2. エリアナビゲーションで、[Protection] エントリを開きます。

可能なアクセスレベルによるテーブルが、インспекタウィンドウに表示されます。

Protection level		Access			Access permission	
Protection		HMI	Read	Write	Password	Confirmation
<input checked="" type="radio"/> Full access (no protection)		✓	✓	✓		
<input type="radio"/> Read access		✓	✓			
<input type="radio"/> HMI access		✓				
<input type="radio"/> No access (complete protection)						

3. テーブルの最初の列で目的の保護レベルを有効にします。それぞれのアクセスレベルの右側の列の緑のチェックマークは、どの操作がパスワードの入力なしで利用できるかを示します。
4. [Password] 列には、選択したアクセスレベルのパスワードを指定します。[Confirmation] 列には、無効なエントリに対して保護するために再度選択したパスワードを入力します。
パスワードが十分セキュアであること、言い換えれば、パスワードが機械で認識できるパターンに従っていないことを確認します。
最初の行にパスワードを入力しなければなりません([Full access] アクセスレベル)。これにより、選択した保護レベルにかかわらず、パスワードを知っているユーザーは CPU に無制限にアクセスできます。
5. 選択済みアクセスレベルで実行できる場合は、必要に応じて、その他のアクセスレベルに追加パスワードを割り当てます。
6. ハードウェアコンフィグレーションを CPU にダウンロードすると、アクセスレベルが有効になります。

動作中のパスワード保護された CPU の動作

CPU 保護は、設定が CPU にダウンロードされた後で有効になります。

オンラインファンクションが実行される前に、必要に応じてアクセス許可がチェックされ、ユーザーはパスワードの入力を求められます。パスワード保護されているファンクションは、1つのプログラミングツール/PCで1回のみ実行できます。別のプログラミングツール/PCはログオンできません。

保護されたデータへのアクセス認証が有効なのは、オンライン接続の間、または[オンライン|アクセス権の削除]でアクセス認証が手動で取り消されるまでです。
パスワード保護された CPU への RUN モード時のアクセスを表示でローカルに制限して、パスワードでのアクセスも行えないようにすることができます。

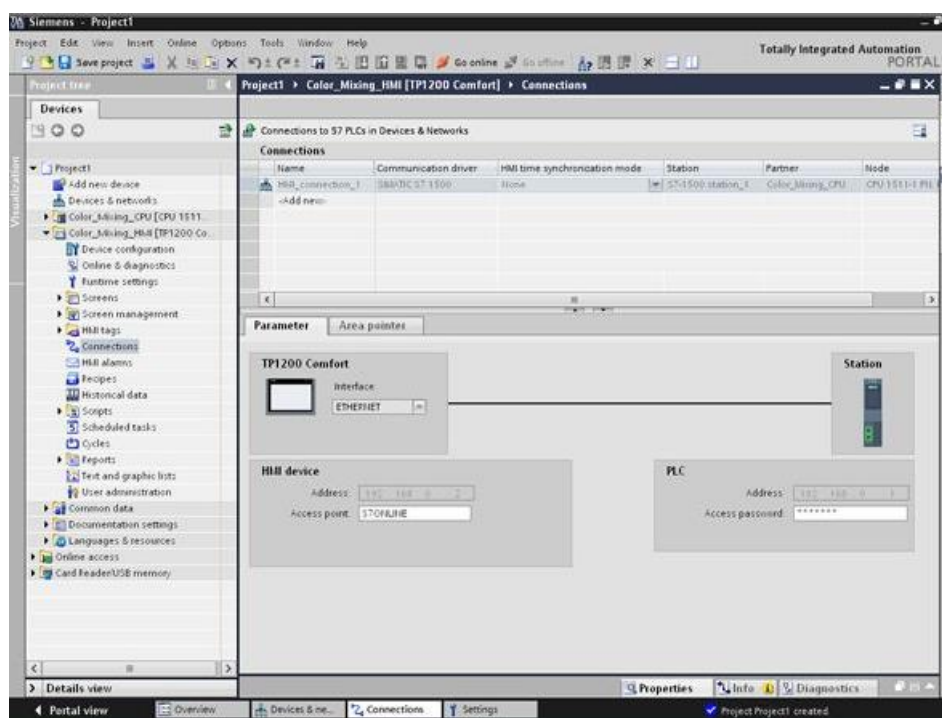
4.7 HMI 接続の保護の設定

概要

保護レベル[Complete protection]が CPU に設定されている場合、HMI デバイスはパスワードがそこに保存されている CPU にだけアクセスできます。
このファンクションは、シーメンスの HMI デバイスでだけ利用できます。

手順

1. プロジェクトツリーで[Connections]エディタを開きます。
2. [Integrated connection]を選択します。
3. [Password]領域に CPU のためのパスワードを入力します。



結果

これで、HMI デバイスは通信が可能で、データを CPU と交換できます。

索引

さ

- システム診断, 109
 - システム診断ウィンドウ, 109
 - システム診断ビュー, 109
 - 診断バッファビュー, 109
 - 詳細ビュー, 109
 - デバイスビュー, 109
 - マトリックスビュー, 111
- システム診断ウィンドウ, 109
- システム診断ビュー, 109

は

- 保護, 131, 133, 134
 - 機械式ロック, 134
 - コピー保護, 133
 - ノウハウプロテクト, 131
 - パスワード保護された CPU の動作, 136

