

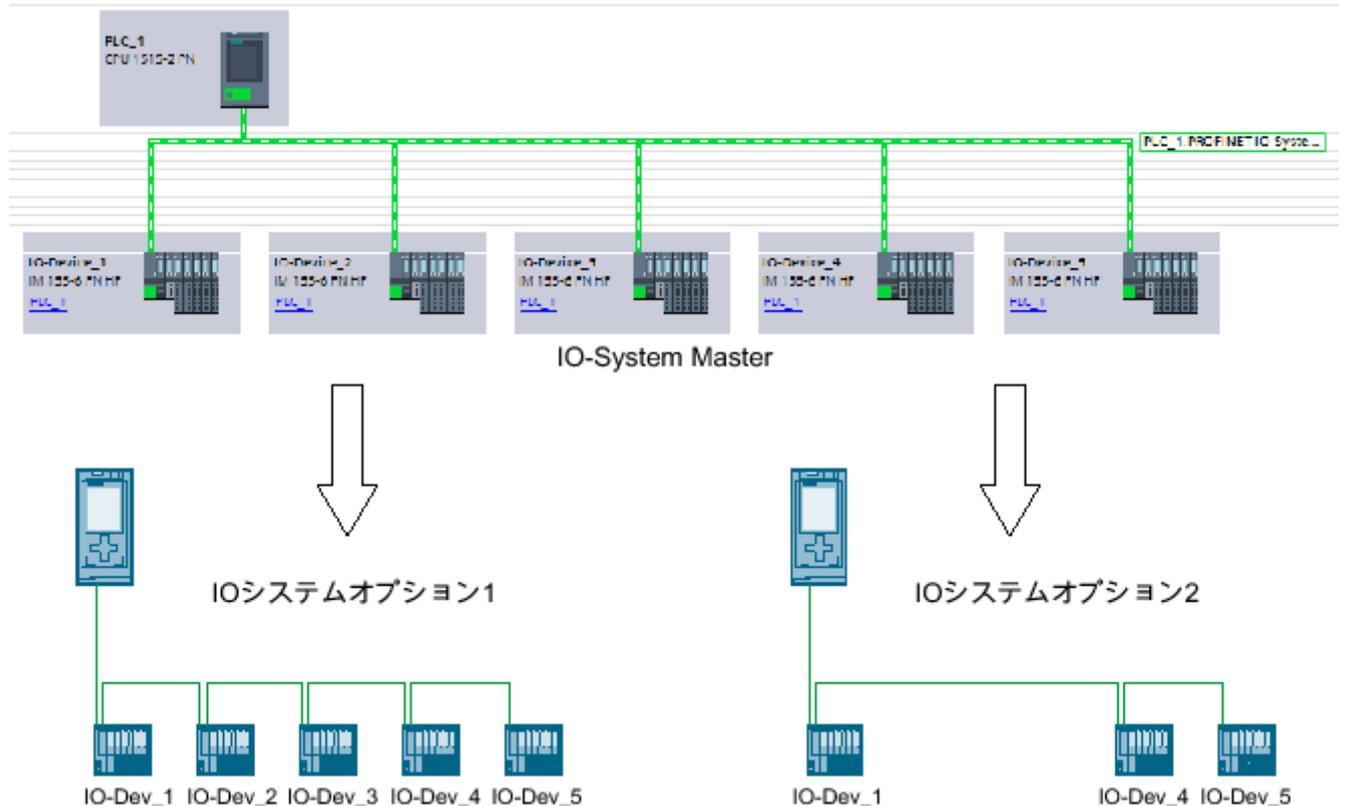
## IO システムの構成制御に関する情報



IO システムの構成制御によって、標準的な機械プロジェクトから標準的な機械の複数の具体的なバージョンを生成することができます。

設定済み構成から実際の構成を取得できる限りは、特定の用途に合わせて IO システムの構成を変更する柔軟性が提供されます。このため、設定された構成は、その構成から取得できるすべての実際の構成の上位セットです。

次の図に、IO システムの数が異なる 2 つの PROFINET IO システムを 1 つの標準的な機械プロジェクトから取得する方法の例を示します。



以降のセクションでは、PROFINET IO システムの構成とプログラミングを実行して、たとえば、構成ソフトウェアを使用しないで、標準的な機械のコミッショニングをオンサイトで行う方法を説明します。

### コンセプト

構成制御の原理は、デバイスレベルでは、サブモジュール/モジュールのフレキシブルな使用(「オプション処理」)のために既に知られています。中央 I/O とリモート I/O の両方に関するさまざまな構成を 1 つのエンジニアリングプロジェクトから取得することができます。

S7-1500 CPU ファームウェアバージョン V1.7 以降では、この原理を IO システムレベルでも適用できます。特定のプラントでは、PROFINET IO システムのステーション(IO デバイス)の順序を省略、追加、または変更するオプションがあります。

デバイス用構成制御と IO システム用構成制御を組み合わせることができますが、ファンクションは相互に独立しています。

IO システムの最大構成とは異なるバリエーションを操作することができます。標準的な機械プロジェクトでは、構成制御を使用して、さまざまな構成に柔軟にカスタマイズ可能な IO デバイスキットを準備できます。

以下のバリエーションが使用できます。

- 関連 IO デバイスの数のバリエーション

必要な構成を備えた適切なデータレコードをユーザープログラムで転送することによって、構成制御用のオプション IO デバイスを構成内に含めます。

- 関連 IO デバイスの順序のバリエーション

必要なトポロジを備えた適切なデータレコードをユーザープログラムで転送することによって、使用するトポロジに合わせて、IO デバイスのポート相互接続を調整します。

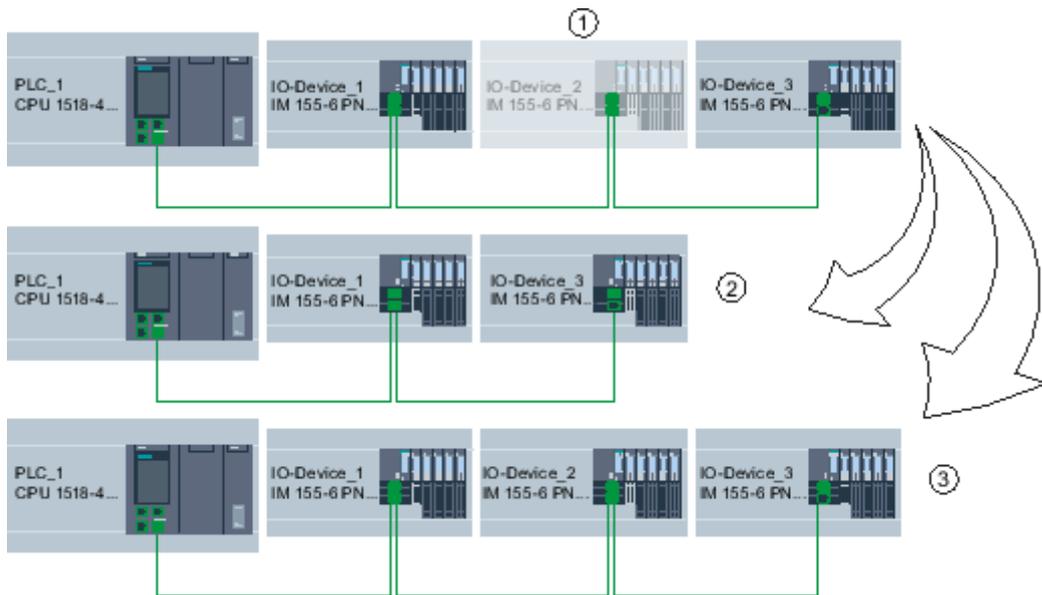
次の図に、1つの IO デバイスが STEP 7 のネットワークビューでオプションとしてマークされている場合に、2つの異なる構成を生成する方法を示します。

- オプション IO デバイスを含まない構成:

この場合、命令[ReconfigIOSystem]を使用して、オプション IO デバイスを構成内に含める必要がないという情報が格納されたデータレコードを PROFINET インターフェースへ転送します。

- オプション IO デバイスを含む構成:

この場合、命令[ReconfigIOSystem]を使用して、オプション IO デバイスを構成に追加するデータレコードを PROFINET インターフェースへ転送します。



- ① パラメータ割り当てによって決定されます: IO Device\_2 はオプション IO デバイスです
- ② オプション IO デバイスを含まない構成
- ③ オプション IO デバイスを含む構成

## 要約: 基本手順

標準的な機械コンセプトの実装時には、以下のフェーズが区別されます。

1. エンジニアリングフェーズ: 標準的な機械プロジェクトの作成と特定のマシンまたはプラントへのロード:

- 特定の機械またはプラントで常に必要なすべての IO デバイス(オプション)の構成の完全な実行
  - 特定の機械またはプラントでは省略されるオプション IO デバイスとしての構成の実行
  - IO デバイスのさまざまな配置に対するポート相互接続の構成の実行
  - スイッチまたは HMI デバイスを使用して、実際に存在する構成をオンサイトで選択することが可能なユーザープログラムの準備
2. コミッショニングフェーズ: 特定の機械またはプラントの操作準備:
- オンサイトネットワークへの機械またはプラントの統合(多目的 IO システムのローカル調整を参照)
  - 設定されたオプションによる、IO システムの現在存在している構成の選択

## オプションとしての IO デバイスの構成の実行



### 必要条件

- S7-1500 CPU ファームウェアバージョン V1.7 以降を IO コントローラとして使用
- STEP 7 V13 SP1 以降
- 標準的な機械プロジェクトの確立と操作の [ルール](#) が考慮されていること。

### 手順

IO デバイスをオプション IO デバイスとして構成定義するには、以下のように実行します。

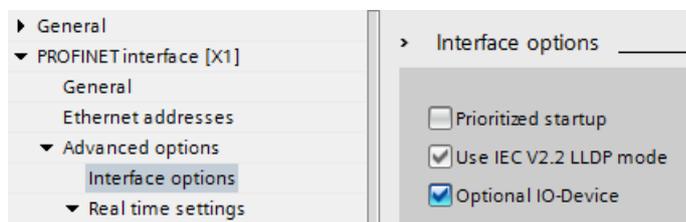
1. プロジェクトを作成します。
2. IO コントローラとして使用する CPU S7-1500 ファームウェアバージョン V1.7 以降を構成定義します。
3. 必要な IO デバイスを構成定義し、それらの IO デバイスを IO コントローラに割り当てます。
4. デバイス間のポート相互接続を構成定義します。

#### 注記

IO システムをユーザープログラムによってカスタマイズするには、IO システムのデバイス間のポート相互接続が必須です。これを行わないと、相互接続された IO デバイスの順序が定義されません。パートナーポートに関しては、以下の設定が可能です:

- IO デバイスの特定のパートナーポート(ポートプロパティの[パートナーポート]ドロップダウンリストによる選択、またはトポロジビューでのドラッグ&ドロップによる相互接続)
- 「ユーザープログラムによるパートナーポートの設定」(ポートプロパティの[パートナーポート]ドロップダウンリストによる選択)。

5. オプションとしてマークを付ける IO デバイスを選択します。
6. 領域[PROFINET インターフェイス[X1]]詳細オプションを選択します。
7. [オプション IO デバイス]オプションを有効にします。
8. オプションとして構成定義するすべての IO デバイスについて、ステップ 5~7 を繰り返します。
9. 構成を CPU にロードします。



**結果:** この構成をロードすると、システムの動作は以下のようになります。

- CPU が、IO システムの構成制御用に準備されます。
- すべての IO デバイスは無効になります。
- その構成をユーザープログラムでカスタマイズするか(オプション IO デバイスを追加する)、またはロードされた構成を変更しないかどうかに関係なく: ユーザープログラムで命令 `ReconfigIOSystem` を呼び出して、現在の構成をシステムに通知する必要があります!

命令 `[ReconfigIOSystem]` を呼び出さないと、システムは操作できません。

この処理のその他の情報は、[プログラムでのオプション IO デバイスの有効化](#)を参照してください。

#### **[I/O 通信]テーブル内の高速パラメータ割り当て**

[I/O 通信]タブで、IO デバイスがオプションであるかどうかを指定することもできます。

追加の[オプション IO デバイス]列で、IO デバイスごとに、IO デバイスがオプションであるかどうかを示す選択可能なチェックボックスを使用できます。ここでは、設定の中央調整が可能です。

## プログラムでのオプション IO デバイスの有効化



### 必要条件

- CPU S7-1500 ファームウェアバージョン V1.7 以降を IO コントローラとして使用。
- STEP 7 V13 SP1。
- 少なくとも 1 つの IO デバイスが、オプション IO デバイスとして構成定義済みであること。
- 標準的な機械プロジェクトの確立と操作の **ルール** が考慮されていること。

### 手順

SIMATIC S7-1500 用ドキュメントに記載されている S7-1500 システムのコミッショニングに関する情報とルールに注意してください!

この手順の次の説明には、オプション IO デバイスのプログラム制御による有効化を理解するために必要なステップのみが含まれています。

オプション IO デバイスを有効/無効にし、これらのデバイスを構成内に含めた後、ユーザープログラムでこれらを有効にするには、以下の手順に従ってください。

1. [ここ](#) の記載に従って、データレコードを作成します。
2. 命令[ReconfigIOSystem]を呼び出し、モード 1 を選択して、すべての IO デバイスを無効にします。

CPU を STOP または POWER OFF 状態に設定して、このステータスでプラントを変更する場合(たとえば、オプション IO デバイスを追加する場合は、モード 1 の[ReconfigIOSystem]による明示的な無効化は不要です。この場合、すなわち、STOP-RUN 移行の後および POWER-OFF > POWER-ON 移行の後に、すべての IO デバイスが自動的に無効になります。

3. リスクなしで再構成が可能な安全なステータスにプラントを移行したときは:

意図した用途に従って、プラントを構成定義します。構成で計画した場所に必要なオプション IO デバイスを追加するか(デバイスの順序を遵守してください!)、不要になったオプション IO デバイスを除去します。

4. IO デバイスのケーブルを配線します。
5. S7-1500 システムをスタートアップして、再度、命令[ReconfigIOSystem]を呼び出します。モード 2 を選択して、データレコード CTRLREC を転送します。
6. データレコードを正常に転送した後に、再度、命令[ReconfigIOSystem]を呼び出します。モード 3 を選択して、現在の構成の一部を形成するすべての IO デバイスを有効にします。

**結果:** CPU は以下の IO デバイスを有効にします。

- オプション IO デバイスとして設定されていないすべての IO デバイス。
- 制御データレコード(CTRLREC)にリストされているすべてのオプション IO デバイス。

以下の IO デバイスは無効なままです。

- ドッキングユニット(操作時に変更中の IO デバイス)。
- 制御データレコードにリストされていないオプション IO デバイス。

### 注記

すべてのモードで同一の制御データレコード(CTRLREC)を使用して、命令[ReconfigIOSystem]を呼び出します!

さまざまなモードでさまざまなデータレコードを使用すると、構成のカスタマイズに不整合が生じ、その結果、かなり理解しにくいエラーメッセージが生成されます。

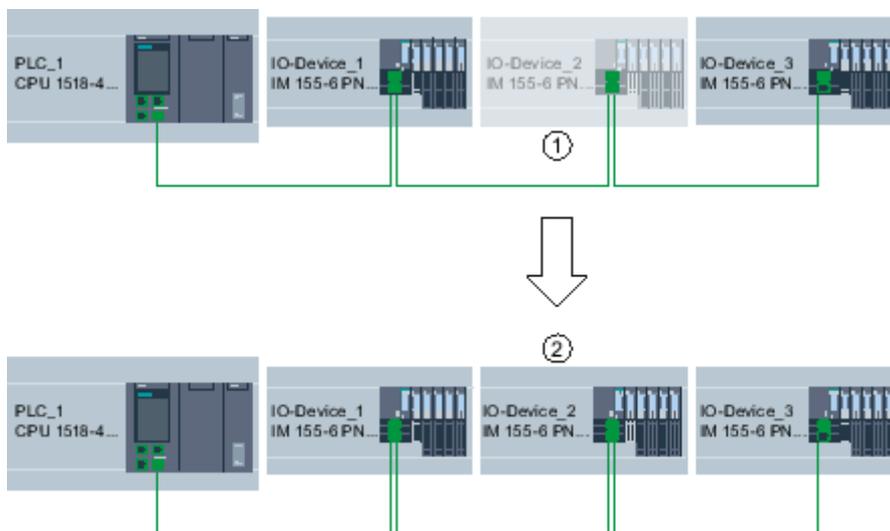
### 例: IO デバイスを有効にするためのデータレコードの構造

ユーザープログラムで、IO デバイス「IO-Device\_2」を唯一の IO デバイスとして有効にする必要があります。

これを行うには、「IO-Device\_2」のハードウェア識別子のみが必要です。

推奨事項: この例で示すように、絶対値の代わりに、ハードウェア識別子のシステム定数を使用してください。この手順では、DB の内容は、構成の変更の結果としてのハードウェア識別子の変更の影響を受けません。

データレコードは、ユーザープログラムで命令[ReconfigIOSystem]を使用して、データブロックに保管し、IO コントローラの PROFINET インターフェースへ転送する必要があります。



- ① IO device\_2 が、オプション IO デバイスとして構成定義されます。
- ② 命令[ReconfigIOSystem]を使用して、データレコードを転送し、構成を有効にすると、IO device\_2 が構成に含められ、IO コントローラとのデータ交換に参加します。

### データブロックの作成

この例では、制御データレコードは 1 つのデータブロック内に作成されます。データブロックは以下の構造を持ちます。

2 行目: 配列の定義 4 つのエレメントを持つタイプ Word の配列。Array of Word は、データタイプとして許容されます。

3 行目: データレコードのバージョン(現在: V1.0)。

4 行目: 有効にするオプション IO デバイスの数(ここでは:1)。

5 行目: IO デバイスのハードウェア識別子のリスト、ここではシステム定数として挿入されます。

6 行目: ユーザープログラムで設定されるポート相互接続の数(ここでは:0)。

7 行目: 追加のデータレコード(オプション)。

myCTRLREC				
	Name	Data type	Start value	Comment
1	Static			
2	ArrMachineConfig0	Array[0..3] of Word		Array for one optional IO device
3	ArrMachineConfig0[0]	Word	16#0100	
4	ArrMachineConfig0[1]	Word	16#0001	
5	ArrMachineConfig0[2]	Word	"IO-Device_2~IODevice"	
6	ArrMachineConfig0[3]	Word	16#0000	
7	ArrMachineConfig1	Array[0..8] of Word		

## [ReconfigIOSystem]の呼び出しシーケンスのルール

- 命令[ReconfigIOSystem]には常に同一の制御データレコード(CTRLREC 入力パラメータ)を提供します!
- POWER OFF -> POWER ON 移行の後の呼び出しシーケンス:
  - モード 1 による ReconfigIOSystem 呼び出し(オプション)。
  - モード 2 による ReconfigIOSystem 呼び出し(必須、以前の再構成がない場合でも!)
  - モード 3 による ReconfigIOSystem 呼び出し(必須)。
- STOP > RUN 移行の後の呼び出しシーケンス:
  - モード 1 による ReconfigIOSystem 呼び出し(オプション)。
  - モード 2 による ReconfigIOSystem 呼び出し(必須、構成が STOP 状態で変更されたときでも)。その他の場合には不要です。
  - モード 3 による ReconfigIOSystem 呼び出し(必須)。
- RUN 状態での再構成用呼び出しシーケンス:
  - モード 1 による ReconfigIOSystem 呼び出し(必須)。
  - モード 2 による ReconfigIOSystem 呼び出し(必須)。
  - モード 3 による ReconfigIOSystem 呼び出し(必須)。

## ルールに関する説明と推奨事項

- オプション IO デバイスとして構成定義する IO デバイスを制御データレコードまたはデータブロック内にリストしないと、この IO デバイスは構成の一部とならず、CPU とのデータ交換に参加しません。
- オプション IO デバイスの有効化を全く行わず、再構成なしのロードされた構成のまま操作する場合は、なお、上記のセクションに記載された手順に従って、制御データレコードを CPU へ転送する必要があります。

制御データレコードは、以下のタグを使用する単純な構造を持ちます。

- バージョン(上位バイト = 1、下位バイト = 0)
- 有効にするオプションデバイスの数 = 0
- ユーザープログラムで設定されるポート相互接続の数 = 0
- STOP-RUN 移行の後および POWER-OFF > POWER-ON 移行の後に、すべての IO デバイスが自動的に無効になります。この理由のため、構成制御を正しく動作させるために、モード 1 による ReconfigIOSystem 呼び出しは必要ありません。

構成制御をプログラミングする汎用的に有効なサンプルとしてユーザーのプロジェクトを使用する場合、すべての再構成の前に、モード 1 による ReconfigIOSystem 呼び出しを実行することをお奨めします。このため、このサンプルは、RUN モードでの再構成でも使用できます。

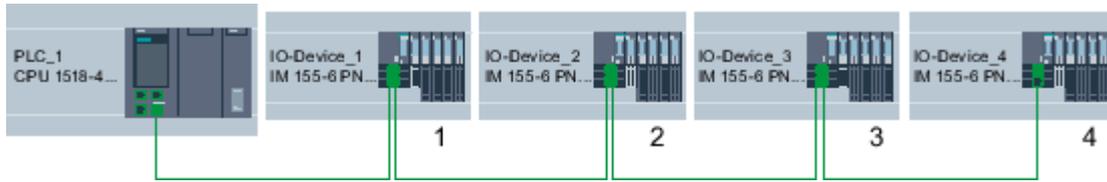
- 同時に IRT を使用している間の、拡張 I/O システム(8 より多くのオプション IO デバイスが存在)のコミッショニング:

オプション IO デバイス(ReconfigIOSystem、モード 3)を有効にするとき、スタートアップ時間を短く保つには、次のヒントに注意してください: IO デバイスのデバイス番号をチェックします。デバイス番号は、IO コントローラから始めてトポロジ相互接続の順序で、昇順に番号付けする必要があります。

ります。IO デバイスがトポロジ上で IO コントローラから遠く離れば離れるほど、すなわち IO コントローラと当の IO デバイスの間に存在する IO デバイスの数が多ければ多いほど、デバイス番号は大きい番号になります。

デバイス番号は、選択した PROFINET インターフェースを備えたインスペクタウィンドウの[イーサネットアドレス - PROFINET]領域で設定します。

線形トポロジによるデバイス番号の割り当ての例:



- CPU は、制御データレコードの非同期転送用の[ReconfigIOSystem]命令を処理します。

このため、スタートアッププログラムでこの命令を呼び出すときは、[BUSY]または[DONE]出力パラメータが、データレコードが転送されたことを示すまで、ループ内で繰り返し[ReconfigIOSystem]を呼び出す必要があります。

ヒント: ループをプログラミングする REPEAT ... UNTIL 命令を備えた SCL プログラミング言語を使用します。

REPEAT

```

    "ReconfigIOSystem"(REQ := "start_config_ctrl",
                       MODE := 1,
                       LADDR := 64,
                       CTRLREC := "myCTRLREC".ArrMachineCon-
fig0,
                       DONE => "conf_DONE",
                       BUSY => "conf_BUSY",
                       ERROR => "conf_ERROR",
                       STATUS => "conf_STATUS");
UNTIL NOT "conf_BUSY"
END_REPEAT;
```

### 追加情報

データレコードの基本構造と、命令[ReconfigIOSystem]の使用については、[ここ](#)を参照してください。