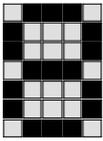


目次

<b>1</b>	<b>診断コンセプトPVIS</b> .....	<b>2</b>
1.1	PSS 4000.....	3
1.1.1	手順 .....	4
1.1.2	基本診断項目の編集.....	7
1.2	PC.....	13
<b>2</b>	<b>ModbusTCP接続: PSS 4000 &lt;-&gt; PMI-Pro</b> .....	<b>15</b>
2.1	PAS4000での設定 .....	15
2.1.1	データフロー: サーバ<->クライアント.....	16
2.1.2	プログラム内のPI変数 (AT %I*およびAT %Q*).....	17
2.1.3	PI変数 (AT) のマッピング .....	18
2.2	PMI-Proでの設定: .....	19
<b>3</b>	<b>Profibus-DP経由でのフィールドバス接続</b> .....	<b>22</b>
3.1	要件.....	22
3.2	Profibus-DP Slave Editor .....	22
3.2.1	Profibus-DPの設定例 .....	24
3.2.2	プログラム内のPI変数 (AT %I*およびAT %Q*).....	25
3.2.3	PI変数 (AT) のマッピング .....	26
3.3	GSDファイル .....	27



## 1 診断コンセプトPVIS

ピルツの診断コンセプト「PVIS」(Pilz visualization) はオペレータおよびプロジェクトコンフィグレータにとって便利なツールで、次の機能を備えています。

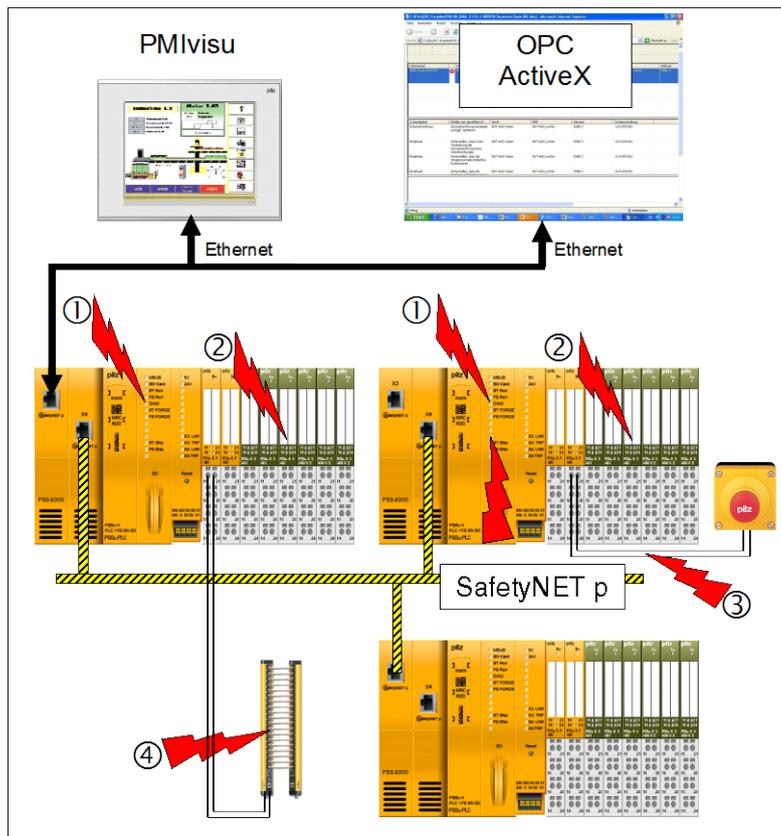
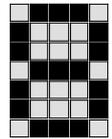
- ☞ プラグアンドプレイ: ピルツのすべてのユニット用のシステムテキストとソリューションの提案は工場出荷時にプリインストールされており、直接使用することができます。
- ☞ エンジニアリング作業の軽減:
  - システム診断  
ハードウェアコンフィグレーションやユーザプログラムのエラー検出を含む、ハードウェアおよびファームウェアの診断です。完全なシステム診断機能を提供し、ユーザが加える調整は、必須のリセットをオフにするなどわずかです。
  - プロセス診断  
手順や状態に関する診断です。プロセス診断は、ユーザによってユーザプログラム内で作成されます。ユーザは、プロセス診断で提供されるピルツコンポーネントブロックを独自の要件に合わせて適応することができます。

メッセージは、次のものを使用して表示することができます。

- ☞ PMLvisu (フルグラフィックタッチスクリーンディスプレイ)
- ☞ PSS 4000 OPC Server (Windows コンピュータ上の PSS 4000 ActiveX Container)

次のエラーまたはオペレータエラーは、PVIS に標準で表示されます。

- ① PSS 4000 の集中型エラーおよび SafetyNET p 経由での分散型エラー
- ② 集中型/分散型入出力モジュールのエラー
- ③ 周辺機器の配線のエラー
- ④ オペレータのエラー (基本診断項目使用時)

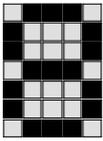


## 1.1 PSS 4000

PSS 4000 のデフォルト設定では、PVIS 機能を使用しません。つまり、イーサネットインターフェースは診断メッセージを送信しません。診断サーバが、プロジェクトの各デバイスからの診断リストと診断ログをまとめて、プロジェクト診断リストとプロジェクト診断ログを構成します。診断サーバは、そのプロジェクト診断リストとログを診断クライアント (PSS 4000 OPC サーバなど) に転送します。その後、診断クライアントがディスプレイ (PSS 4000 ActiveX Control など) に情報を提供します。

プロジェクト診断リストとプロジェクト診断ログは PAS4000 にも表示されます。デバイスの診断リストと診断ログは表示されません。

ユーザは PAS4000 ソフトウェアを使用して、わずかな手順を実行するだけで PVIS 機能を有効にできます。

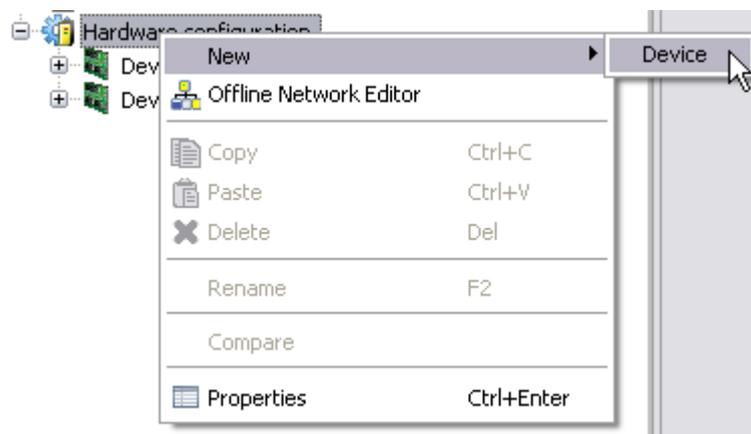


### 1.1.1 手順

#### 1.1.1.1 手順 1: 新規デバイスの追加

最初の手順では、OPC Server への接続装置として機能する新規デバイスをハードウェアコンフィグレーションに追加します。

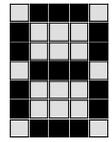
- ① プロジェクトマネージャの [Hardware Configuration] で、コンテキストメニュー [New] → [Device] から新規デバイスを選択します。



- ② 製品タイプを選択します。この場合は [PSS 4000-PC] です。



- ③  ボタンをクリックして、設定を続行します。
- ③ [Device Name] はデフォルトの名前 (例: 「NB-KDELEON」) をそのまま使用します。



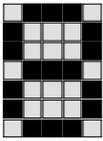
- ④ [IP address] に、OPC Server の IP アドレス (例: 「192.168.1.2」) を指定します。



[Serial number] の入力は不要です。 **Finish** をクリックして、ハードウェアコンフィグレーションを完了します。



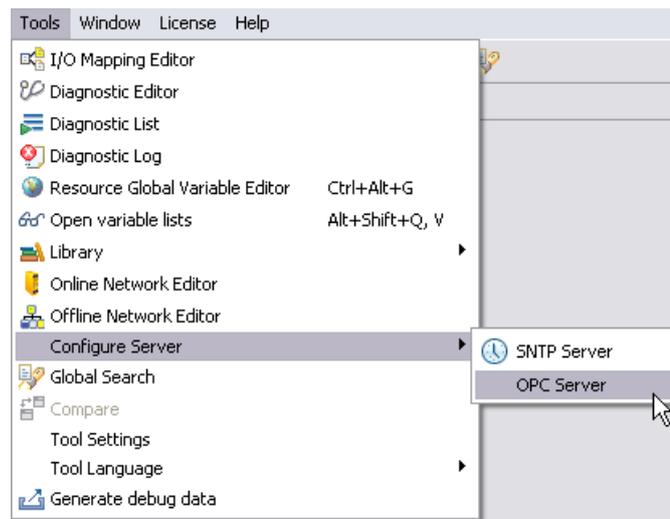
プロジェクトマネージャに新しいデバイスが表示されます。



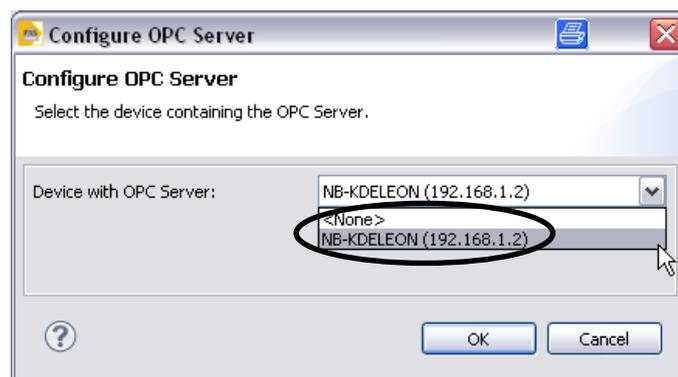
## 1.1.1.2 手順 2: デバイスの割り付け

次に、デバイス「myPC」を OPC Server としてコンフィグレーションする必要があります。

- 1 コンフィグレーションは、[Tools] → [Configure Server] → [OPC Server] から行います。



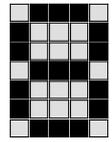
- 2 OPC Server としてコンフィグレーションするデバイスを選択します。  
[myPC (192.168.1.110)]



- 3  をクリックして確定します。

 これで PSS 4000 での準備が完了し、PSS 4000 に PVIS 機能が装備されました。

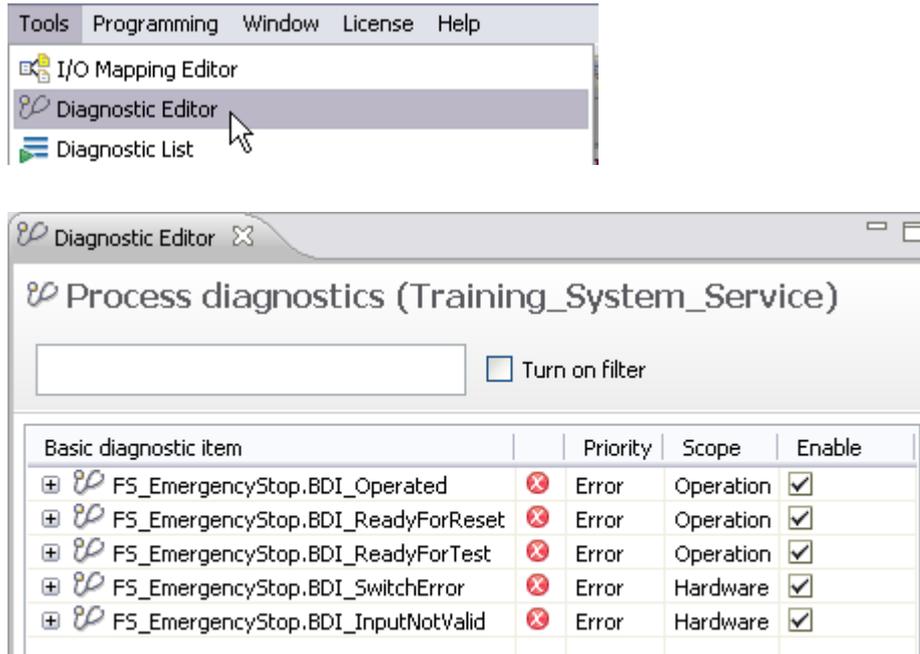
次に、プロジェクトを構築 → ダウンロードする必要があります。



### 1.1.2 基本診断項目の編集

ユーザは、プロセス診断で提供されるピルツコンポーネントブロックを独自の要件に合わせて適応することができます。また、独自の基本診断項目を作成することもできます。

- ① 診断タイプを編集するには、[Tools] → [Diagnostic Editor] を選択します。

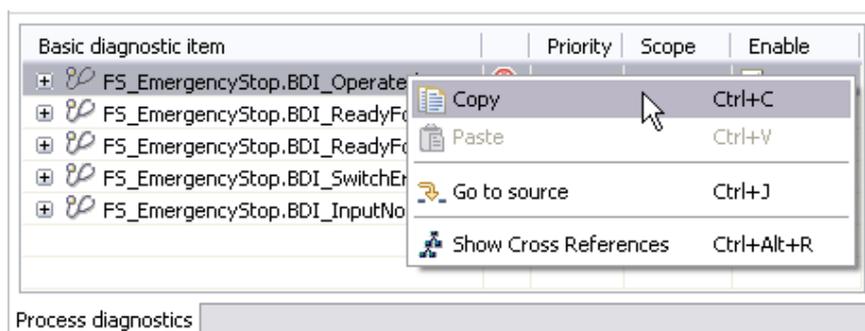


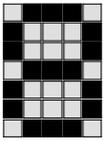
基本診断項目を編集するプログラムが有効になっている必要があります。プロジェクトツリーで、プロジェクトのフォルダに緑色の記号が表示されていることが必要です →

#### 1.1.2.1 手順 1: 保護のための基本診断項目の変更

保護のための新しい基本診断項目 (BDI) を作成するには、まず既存の基本診断項目をコピーする必要があります。

- ① 変更する基本診断項目を右クリックして、コンテキストメニューから [Copy] を選択します。

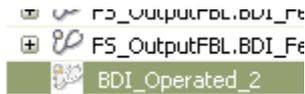




完全に一致する基本診断項目のコピーが作成されます。必要に応じて名前を変更できます。



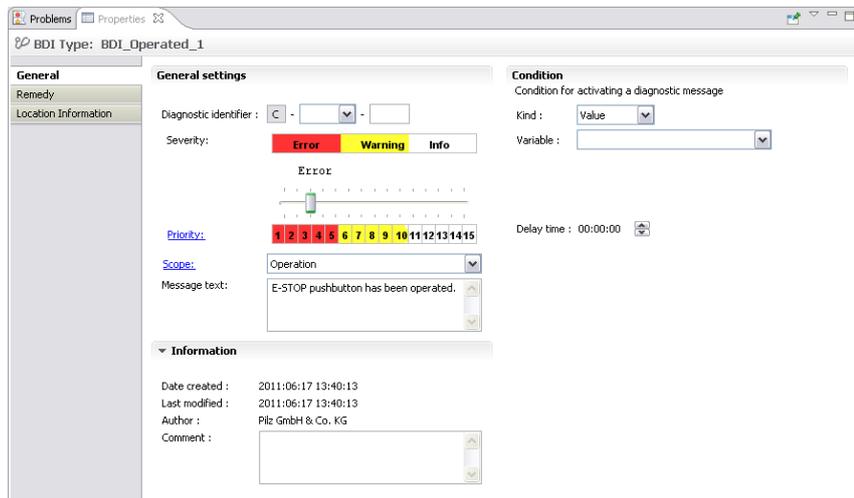
② 基本診断項目を選択して、プロパティを開きます。



情報領域の新しいタブにプロパティが表示されます。

③ [Properties] タブには、さらに3つのタブが含まれています。

### 1. [General]



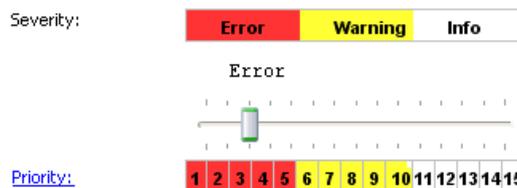
### 2. [General settings]



診断 ID は、診断リストの診断メッセージ、または診断ログのログエントリで確認できます。

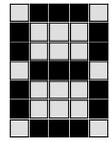


優先度はスライドコントロールで変更できます。

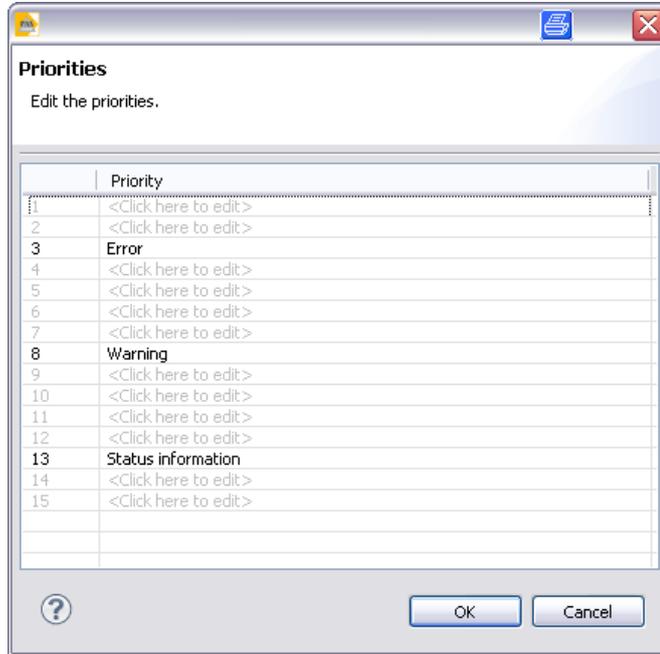




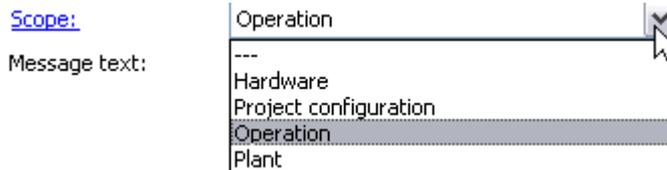
## PVIS (Pilz Visualization)



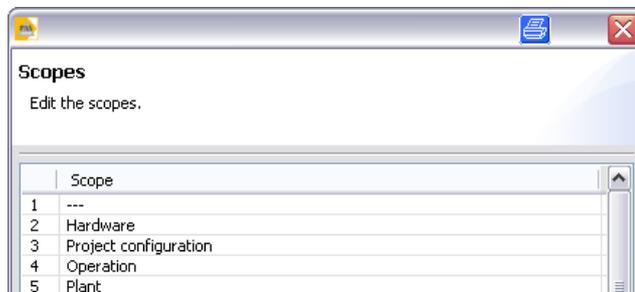
- ④ テーブルに「Test123」などのテキストを入力して、ハイパーリンク「[Priorität:](#)」で呼び出します。



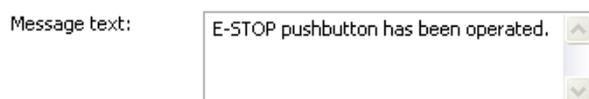
- ⑤ 対象はコンボボックスで選択できます。

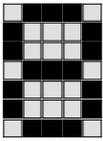


コンボボックス内のテキストは、ハイパーリンク「[Scope:](#)」で変更することができます。



メッセージテキストはテキストフィールドで変更できます。





• [Information]

**Information**

Date created : 2011:06:17 13:40:13  
 Last modified : 2011:06:17 13:40:13  
 Author : Pilz GmbH & Co. KG  
 Comment :

• [Condition]

**Condition**  
 Condition for activating a diagnostic message

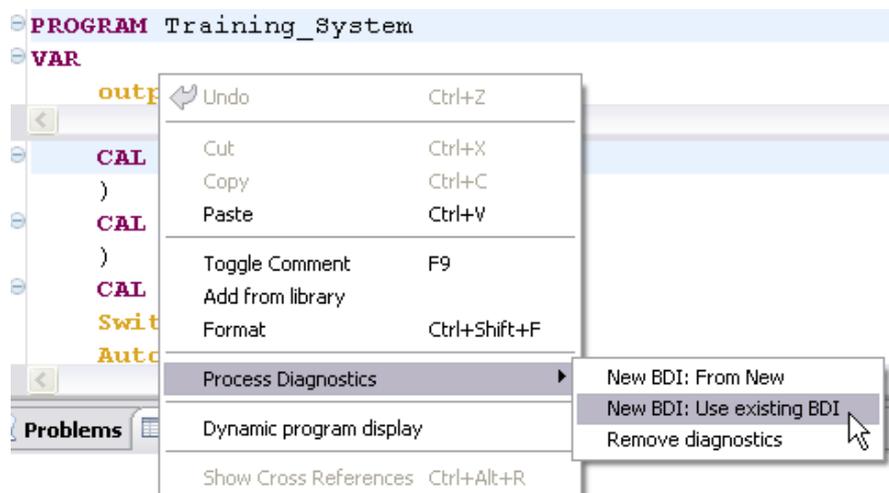
Kind : Value   
 Variable :

Delay time : 00:00:00



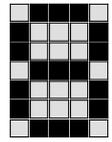
[Condition] は、変数を指定して直接 BDI を割り付けるまで使用可能になりません。

- ⑥ 変数のコンテキストメニューから [Process Diagnostic] → [New BDI:Use existing BDI] を選択します。

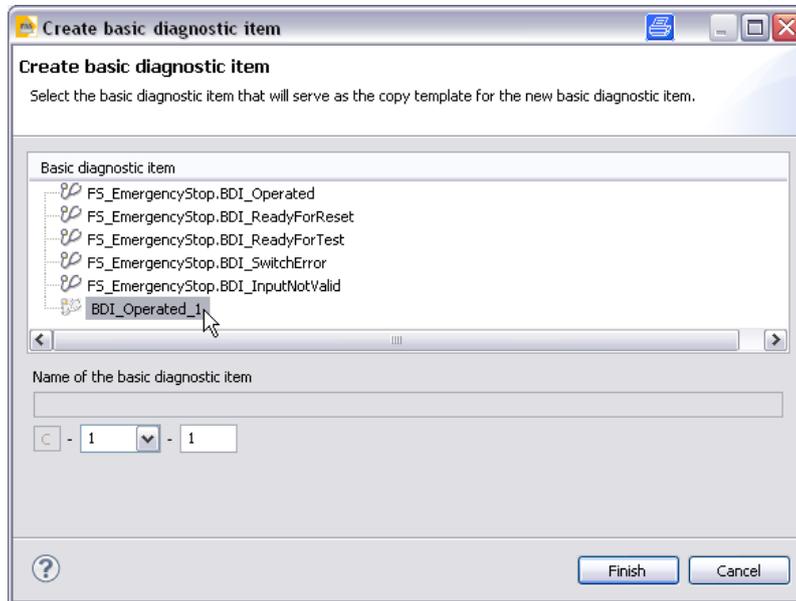


o object has been selected for which the properties are to be displayed.





- ⑦ 作成された BDI を選択します。
  - BDI の診断 ID は「C - 1005 - 1」になっています。

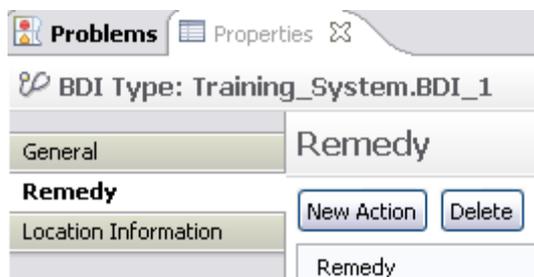


- ⑧ **Beenden** を押して手順を完了します。

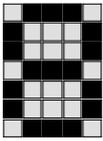
1. ステータスの変化: `var1: BOOL;` → `var1: BOOL;`

### a. [Remedy]

- ☞ **New Action** ボタンを使用して他のアクションをリストに追加したり、  
**Delete** ボタンを使用してアクションをリストから削除できます。



アクションを選択して、編集することもできます。



Action text:

Scope :

Responsibility :

Location information

Kind:

- ① [Action Text] フィールドにテキストを入力します。
- ② 対象はコンボボックスで選択し、ハイパーリンクを使って編集できます。
- ③ 担当者はコンボボックスで選択し、ハイパーリンクを使って編集できます。
- ④ 位置情報はコンボボックスで選択できます。

b. [Location information]



位置情報の種類は、コンボボックスで指定できます。

Problems Properties

BDI Type: Training\_System.BDI\_1

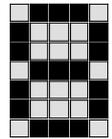
General

Remedy

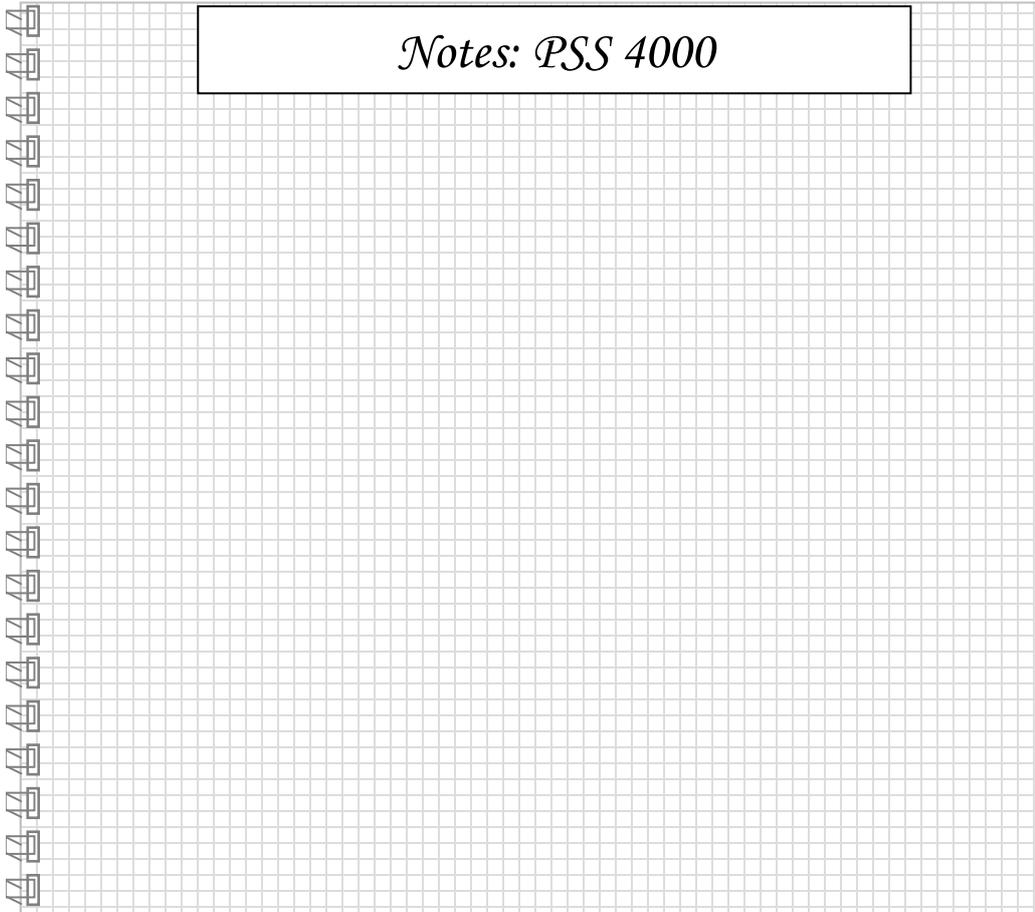
Location Information

Kind:

- Sensor/actuator
- Terminal
- Sensor/actuator
- POU
- User-defined



*Notes: PSS 4000*



## 1.2 PC

「PVIS-OPC Server」および「PVIS-ActiveX Container」を使用することで、非常に少ないエンジニアリング作業量で PVIS のすべての機能を実装することができます。「PVIS-ActiveX Container」は、プレーンテキストメッセージとソリューションの提案を次のものから提供します。

- PSS 4000 システムメッセージ
- 保護 (BDI) のメッセージ
- PSS 4000 オペランド

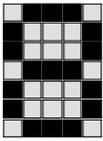
「PVIS OPC Server」はイーサネットカード (LAN) インタフェースにアクセスし、PSS 4000 内のすべての定義済みイベントを記録して、「PVIS-ActiveX Container」がデータを使用できるようにします。



### HTMLによるActiveX Containerの表示

ActiveX Container は、HTML ファイルを使用して Internet Explorer で表示することで、簡単に表示できます。



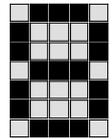


The screenshot shows the PVIS software interface. The title bar indicates the project is 'Projekt: Training\_System\_Service' and the window title is 'Diagnoseliste (1 / 4)'. The main area displays a list of diagnostic messages:

		Info (F2)	Abhilfe (ENT)	Param. (ESC)
	Error			
<b>DID:</b>	17.06.2011 13:56:55			
<b>Bereich:</b>	P-0001-0001			
<b>Gerät:</b>	Operation			
<b>BMK:</b>	---			
<b>Ortsbeschr.:</b>	---			
<b>Adresse:</b>	---			
<b>Beschreibung:</b>	E-STOP pushbutton has been operated.			
	Status infor...			
<b>DID:</b>	17.06.2011 13:36:25			
<b>Bereich:</b>	S-0303-0001			
<b>Gerät:</b>	Project configuration			
<b>BMK:</b>	---			
<b>Ortsbeschr.:</b>	---			
<b>Adresse:</b>	Device_1			
<b>Beschreibung:</b>	Project is unlicensed.			
	Status infor...			
<b>DID:</b>	17.06.2011 13:36:16			
<b>Bereich:</b>	S-0303-0001			
<b>Gerät:</b>	Project configuration			
<b>BMK:</b>	---			
<b>Ortsbeschr.:</b>	---			
<b>Adresse:</b>	Device_0			
<b>Beschreibung:</b>	Project is unlicensed.			

The status bar at the bottom shows 'Fertig' and 'Eigener Computer'.

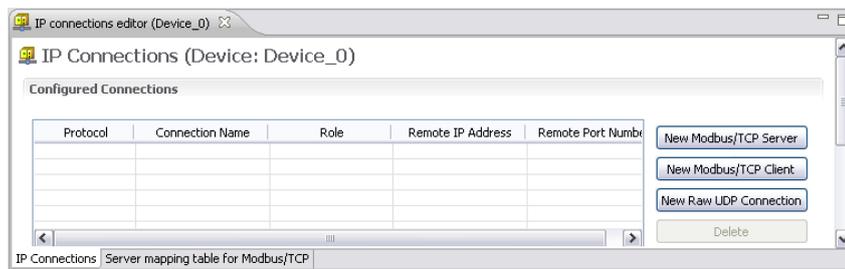
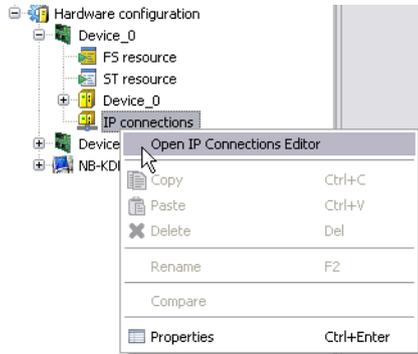




## 2 ModbusTCP接続: PSS 4000 <-> PMI-Pro

### 2.1 PAS4000 での設定

- ① 次の設定で、新しい IP 接続をセットアップします。

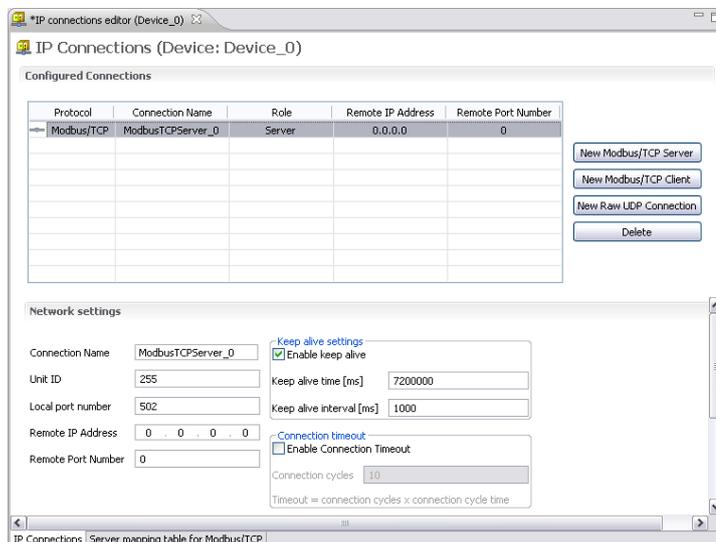


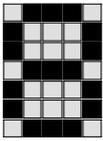
- ② PSS 4000 は、Modbus/TCP サーバまたは Modbus/TCP クライアントとしてコンフィグレーションできます。



通常、PSS 4000 は、Modbus/TCP サーバとしてコンフィグレーションします。

- ③ 新しい Modbus/TCP サーバをセットアップするには、New Modbus/TCP Server ボタンをクリックします。



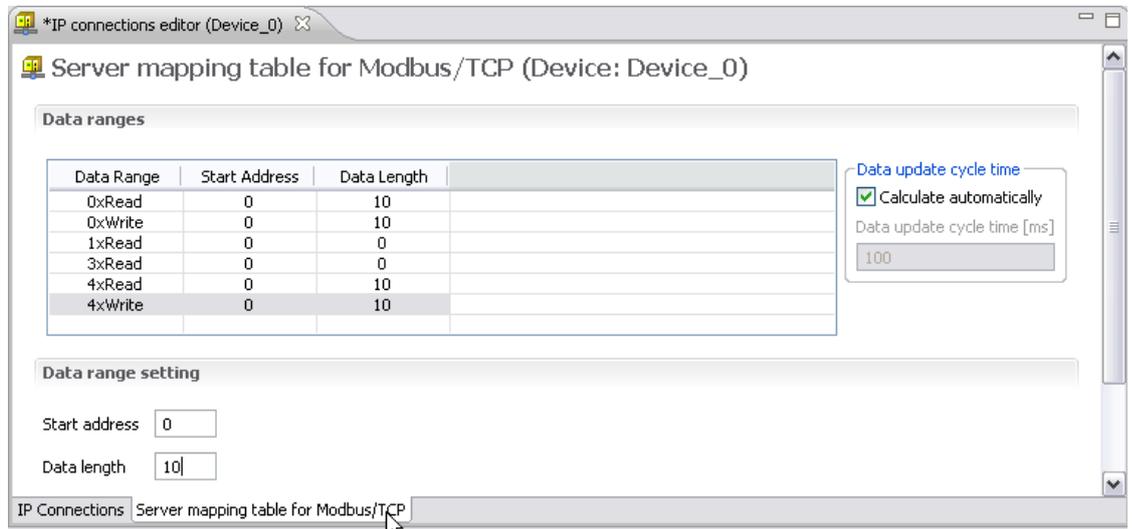


④ 接続名と IP アドレスを入力します。

☞ 設定「Remote-IP-Adresse 0.0.0.0」と「Remote-Port-Nummer 0」は、一斉通信として機能します。この場合、ネットワーク内のすべてのリモート装置が通信内容を記録して、それぞれに宛てられたすべてのデータをフィルタリングします。

⑤ 次のデータ範囲のデータの長さは、IP Connections Editor で設定できます。

- 0xRead -> BOOL (コイル)
- 0xWrite -> BOOL (コイル)
- 1xRead -> BOOL (入力レジスタ)
- 3xRead -> WORD (入力レジスタ)
- 4xRead -> WORD (保持レジスタ)
- 4xWrite -> WORD (保持レジスタ)



☞ 各変数が PMI-Pro によって変更されるようにするには、PAS4000 に入出力変数が必要です。

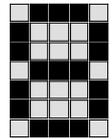
### 2.1.1 データフロー: サーバ <-> クライアント

データフロー、つまり [Data Range] 列の READ/WRITE 設定は、クライアントからの視点です。

#### 1. PSS 4000 がサーバの場合 (推奨設定)

データ範囲	データフロー	PI 変数
READ	データの方向: PSS 4000 (サーバ) -> クライアント	出力変数%Q*
WRITE	データの方向: クライアント -> PSS 4000 (サーバ)	入力変数%I*





## 2. PSS 4000 がクライアントの場合

データ範囲	データフロー	PI 変数
WRITE	データの方向: PSS 4000 (クライアント) -> サーバ	出力変数%Q*
READ	データの方向: サーバ-> PSS 4000 (クライアント)	入力変数%I*



クライアントによって読み取られた入力変数 (%I\*) は、PSS 4000 プログラムでミラーリングされる必要があります。

### 2.1.2 プログラム内のPI変数 (AT %I\*およびAT %Q\*)

```

PROGRAM Modbus
VAR
I_b_input_safebool AT %I*:SAFEBOOL;           //Hardware Input 1
Q_b_output_safebool AT %Q*:SAFEBOOL;           //Hardware Output 1
Q_b_output_safebool_2 AT %Q*:SAFEBOOL;         //Hardware Output 2

I_w_input_data_word AT %I*:WORD;               //WORD Variable written by PMI-Pro
Q_w_output_data_word AT %Q*:WORD;             //WORD Variable read by PMI-Pro

I_b_input_data_bool AT %I*:BOOL;              //BOOL Variable written by PMI-Pro
Q_b_output_data_bool AT %Q*:BOOL;             //BOOL Variable read by PMI-Pro

I_b_input_data_bool_2 AT %I*:BOOL;            //BOOL Variable written by PMI-Pro
Q_b_output_data_bool_2 AT %Q*:BOOL;           //BOOL Variable read by PMI-Pro

Q_b_output_data_bool_3 AT %Q*:BOOL;           //BOOL Variable read by PMI-Pro
END_VAR

(*PMI-Pro able to change WORD-Variables*)
LD      I_w_input_data_word
ST      Q_w_output_data_word

(*PMI-Pro able to change BOOL-Variables *)
LD      I_b_input_data_bool
ST      Q_b_output_data_bool

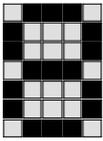
(*PMI-Pro able to set Hardware-Output 1*)
LD      I_b_input_data_bool_2
EQ      TRUE
ST      Q_b_output_safebool
ST      Q_b_output_data_bool_2 // PMI-Pro read Status
                                     of Hardware-Output 1

(*PMI-Pro able to set Hardware-Output 1*)
LD      I_b_input_safebool
EQ      TRUE
ST      Q_b_output_safebool_2
ST      Q_b_output_data_bool_3 // PMI-Pro read Status
                                     of Hardware-Output 2

END_PROGRAM

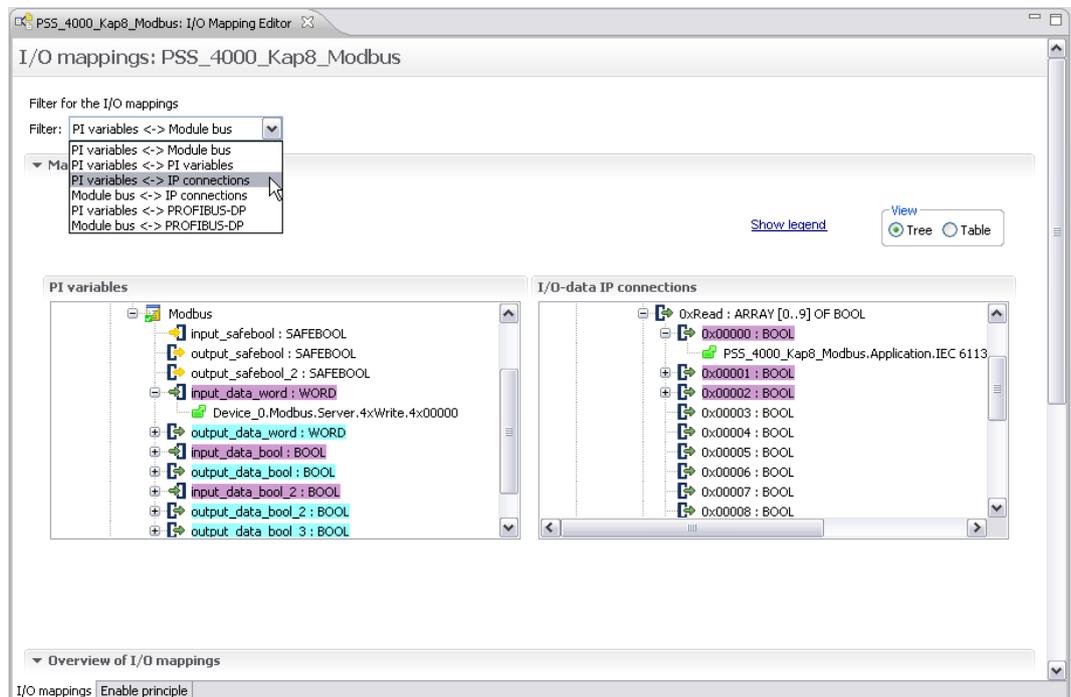
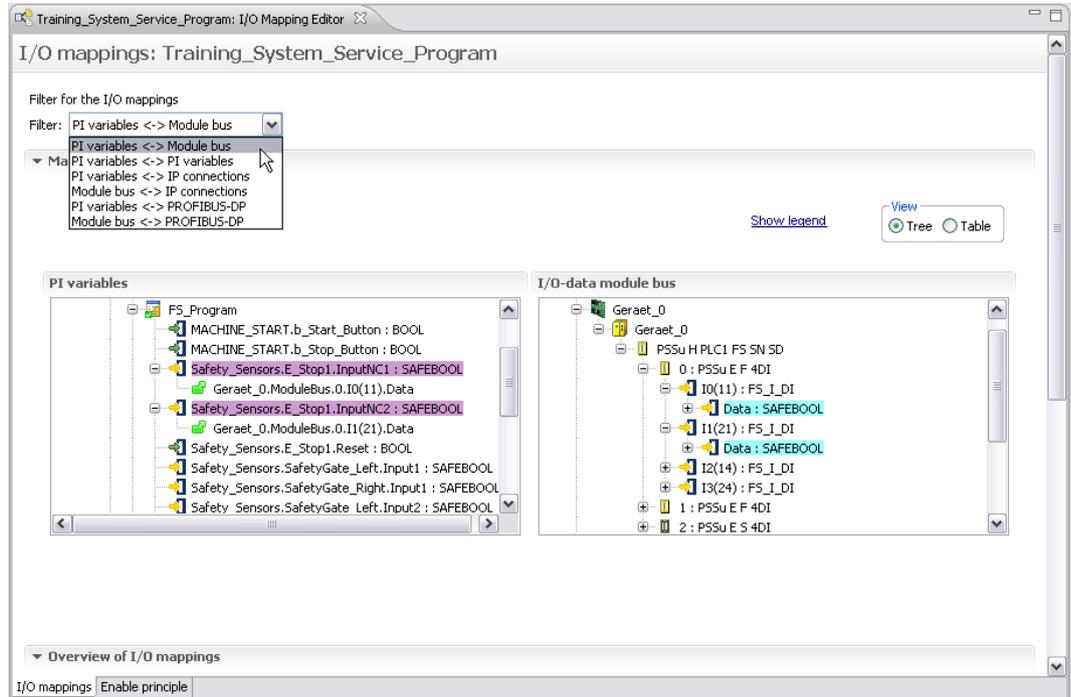
```

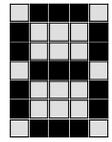




### 2.1.3 PI変数 (AT) のマッピング

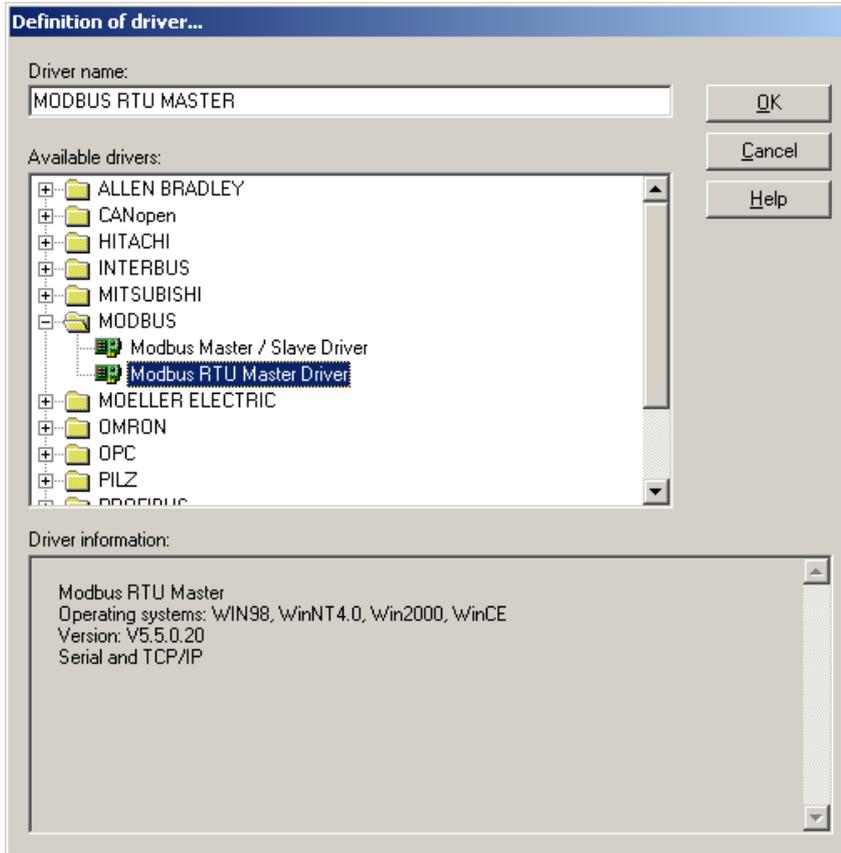
コードワード「AT」(入力変数および出力変数)のPI変数は、I/O-Mapping Editorでマップする必要があります。



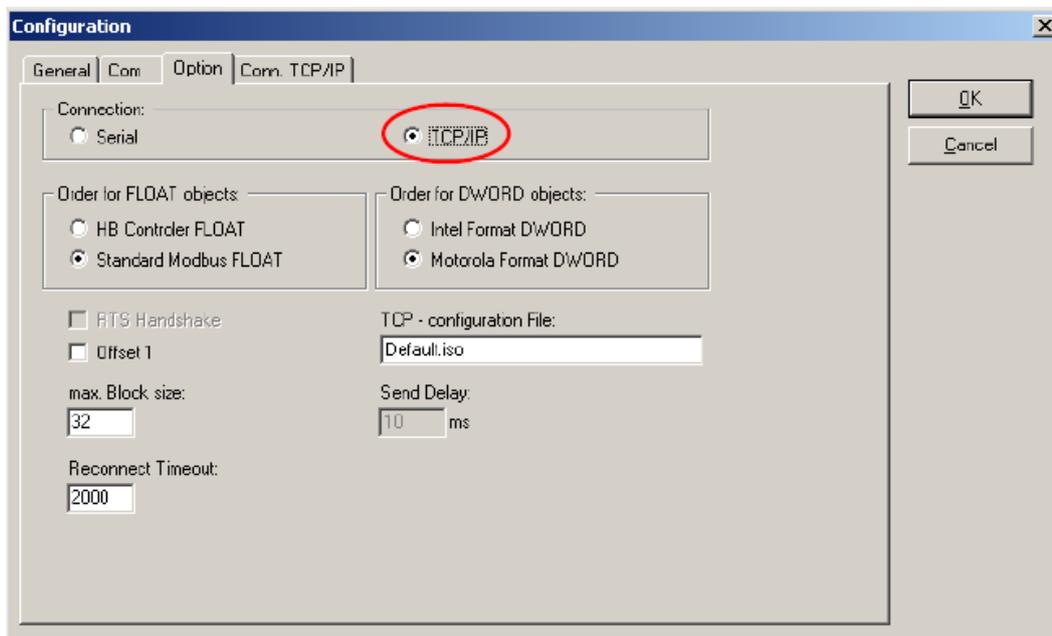


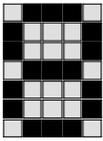
## 2.2 PMI-Proでの設定:

- ① 新しい Modbus ドライバ [Modbus RTU Master Driver] を選択します



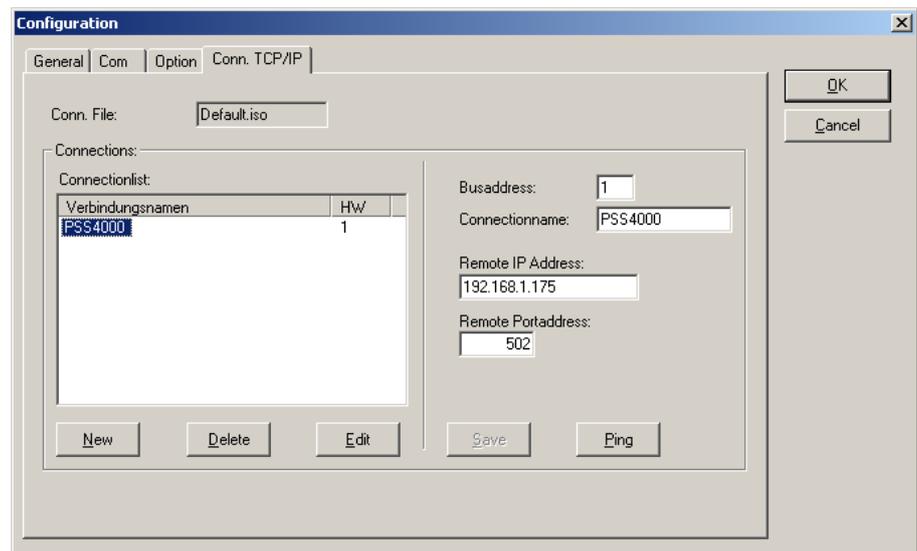
- ② [Option] タブで TCP/IP 接続ドライバを設定します。



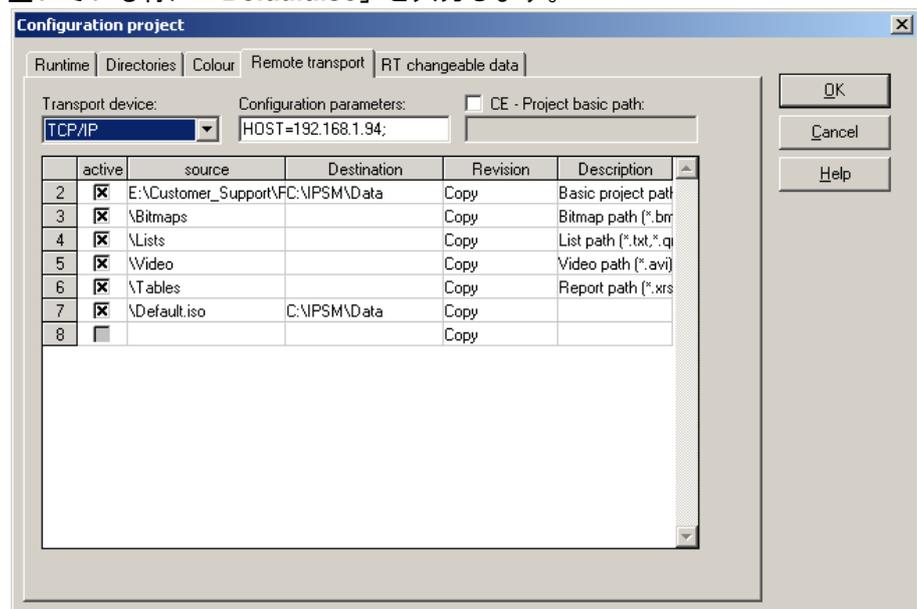


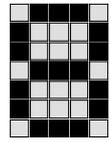
PMI-Pro が default.iso 接続ファイルを生成し、これを Project フォルダに保存します。

- ③ [Conn. TCP/IP] タブに新しい TCP/IP 接続を追加し、リモート IP アドレスを指定します。  
たとえば、次のリモート IP アドレスを指定します。  
Geraet\_0 の IP: 192.168.1.100  
Geraet\_1 の IP: 192.168.1.101

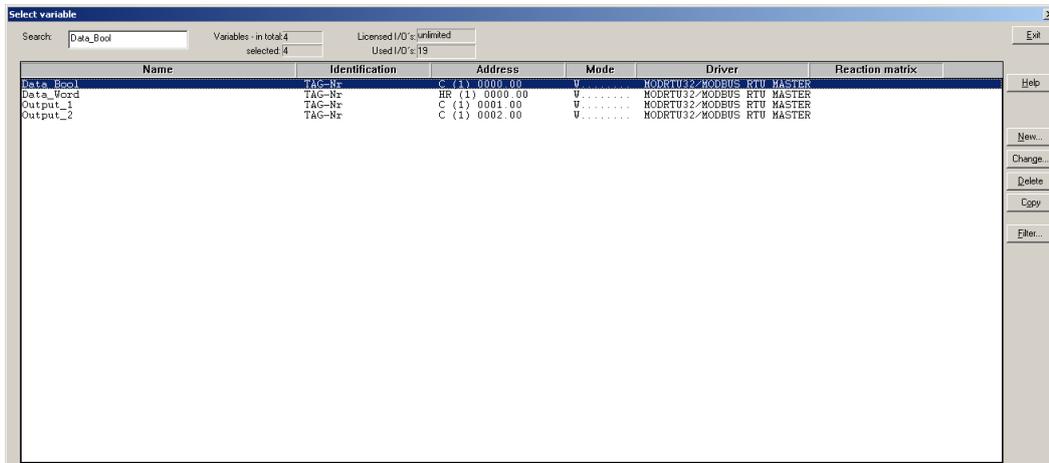


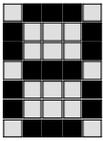
- ④ 宛先フォルダ C:\IPSM\Data を指定して、Default.iso を [Remote Transport] に指定する必要があります。  
メニューから [File] ⇒ [Project configuration] ⇒ [Project...] を選択し、空いている行に「Default.iso」と入力します。





- ⑤ 変数を読み取れるように、変数リストに追加する必要があります。

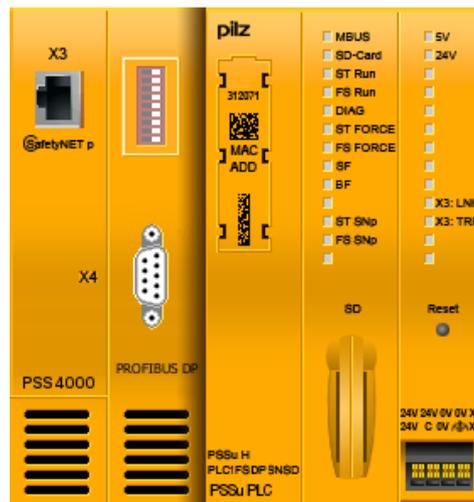




### 3 Profibus-DP経由でのフィールドバス接続

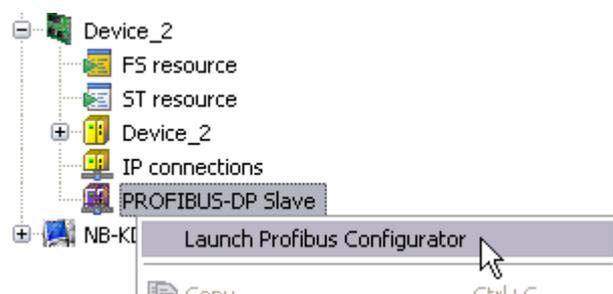
#### 3.1 要件

Profibus-DP 経由でフィールドバス接続を行うには、ヘッドモジュール「PSSu H PLC1 FS DP SN SD」が必要です。



#### 3.2 Profibus-DP Slave Editor

- 1 [Project Management] ペインで適切な項目をクリックして、DP Slave Editor を開きます。

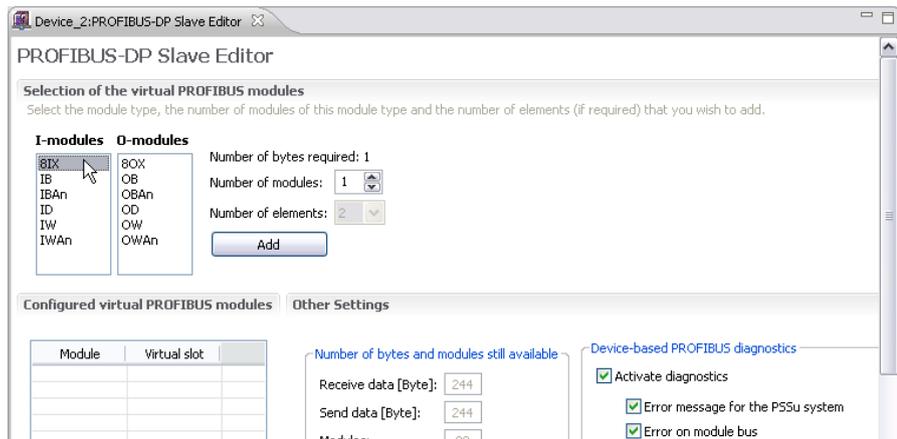
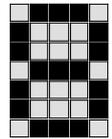


PSSu システムでは、必要な仮想 PROFIBUS モジュールは PROFIBUS-DP Slave Configurationによってコンフィグレーションされます。仮想 PROFIBUS モジュールは、モジュールをコンフィグレーションした後に仮想スロット (0 … 79) で確認できます。

PSSu システムから見た場合、受信情報は外部入力データで、発信情報は外部出力データです。



# PVIS (Pilz Visualization)



② 次の I/O モジュールを実装できます。

- I-modules**
- 8IX
  - IB
  - IBAn
  - ID
  - IW
  - IWAn

入力モジュール	意味	変数の宣言
8IX	バイナリ入力 x 8	AT %I* : BOOL (8 回)
IB	バイト入力 x 1	AT %I* : BYTE
IW	ワード入力 x 1	AT %I* : WORD
ID	ダブルワード入力 x 1	AT %I* : DWORD
IBAn	入力配列 (バイト形式) x 1	AT %I* : ARRAY [0..63] OF BYTE
IWAn	入力配列 (ワード形式) x 1	AT %I* : ARRAY [0..63] OF WORD



入力モジュールは入力データバッファとして機能します。Profibus-Master の出力データを受信します。

- O-modules**
- 8OX
  - OB
  - OBAn
  - OD
  - OW
  - OWAn

出力モジュール	意味	変数の宣言
8OX	バイナリ出力 x 8	AT %Q* : BOOL (8 回)
OB	バイト出力 x 1	AT %Q* : BYTE
OW	ワード出力 x 1	AT %Q* : WORD
OD	ダブルワード出力 x 1	AT %Q* : DWORD
OBAn	出力配列 (バイト形式) x 1	AT %Q* : ARRAY [0..63] OF BYTE
OWAn	出力配列 (ワード形式) x 1	AT %Q* : ARRAY [0..63] OF WORD



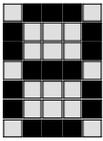
出力モジュールは出力データバッファとして機能します。Profibus-Master への入力データを送信します。

③ 次に、モジュールおよびファンクション (配列の場合のみ) の数を指定します。これらの値は、使用するモジュールの数を指定します。

Number of bytes required: 16  
 Number of modules:   
 Number of elements:

入力モジュール	出力モジュール	モジュール数	ファンクション数
8IX	8OX	1 ... 80	
IB	OB	1 ... 80	





IW	OW	1 ... 80	指定不可
ID	OD	1 ... 61	
IBAn	OBAn	1 ... 80*	2, 4, 8, 16, 32, 64*
IWAn	OWAn	1 ... 80*	2, 4, 8, 16, 32, 64*

\* 配列の場合、モジュール数がファンクション数で乗算されます。この値が 244 バイトを超えてはいけません。

### 3.2.1 Profibus-DPの設定例

- ① Profibus-DP 経由の通信では次のデータ長を使用する必要があります。

受信データ長 = 24 バイト。内訳は以下の通り。			
バイト	変数のデータ型	入力モジュール	モジュール数／ファンクション数
16	BOOL	8IX	16/-
8	WORD	IWAn	1/4
送信データ長 = 32 バイト。内訳は以下の通り。			
バイト	変数のデータ型	入力モジュール	モジュール数／ファンクション数
16	BOOL	8IX	16/-
16	WORD	IWAn	1/8

- ② 入力モジュール [8IX] を選択して、モジュール数を 16 に設定します。

をクリックして確定します。

**I-modules**   **O-modules**

8IX	8OX
IB	OB
IBAn	OBAn
ID	OD
IW	OW
IWAn	OWAn

Number of bytes required: 16

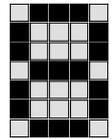
Number of modules:

Number of elements:

**Configured virtual PROFIBUS modules**

Module	Virtual slot
8IX	4
8IX	5
8IX	6
8IX	7
8IX	8
8IX	9
8IX	10
8IX	11
8IX	12
8IX	13
8IX	14
8IX	15





- ③ 入力モジュール [IWA<sub>n</sub>] を選択します。モジュール数を1、ファンクション数を4に設定します。 **Hinzufügen** をクリックして設定を確定します。

**I-modules**   **O-modules**

8IX	8OX	Number of bytes required: 8
IB	OB	Number of modules: 1
IBAn	OBAn	Number of elements: 4
ID	OD	<b>Add</b>
IW	OW	
<b>IWA<sub>n</sub></b>	<b>OWAn</b>	

8IX	15
<b>IWA4</b>	<b>16</b>

- ④ 送信データについて手順 ② と ③ を繰り返します (出力モジュールの場合は [8OX] と [OWAn])。

**Configured virtual PROFIBUS modules**

Module	Virtual slot
8OX	22
8OX	23
8OX	24
8OX	25
8OX	26
8OX	27
8OX	28
8OX	29
8OX	30
8OX	31
8OX	32
<b>OWA4</b>	<b>33</b>

**Delete**

### 3.2.2 プログラム内のPI変数 (AT %I\*およびAT %Q\*)

```

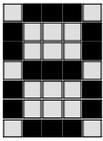
VAR

I_b_myFirstProfibus_Input0 AT %I*:BOOL;
I_b_myFirstProfibus_Input1 AT %I*:BOOL;
I_b_myFirstProfibus_Input2 AT %I*:BOOL;
...
I_b_myFirstProfibus_Input127 AT %I*:BOOL;

Q_b_myFirstProfibus_Output0 AT %Q*:BOOL;
Q_b_myFirstProfibus_Output1 AT %Q*:BOOL;
Q_b_myFirstProfibus_Output2 AT %Q*:BOOL;
...
Q_b_myFirstProfibus_Output127 AT %Q*:BOOL;

END_VAR
    
```





### 3.2.3 PI変数 (AT) のマッピング

キーワード「AT」の PI 変数をマップします (入力変数と出力変数は、I/O Mapping Editor で、I/O モジュールとその仮想スロット間にマップする必要があります)。

**Mapping Editor**

**Profibus-DP Slave Editor  
I/Oモジュール**

**POU内のPI変数  
AT %I\*およびAT %Q\***

```

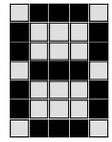
VAR
I_b_myFirstProfibus_Input0 AT %I*:BOOL;
I_b_myFirstProfibus_Input1 AT %I*:BOOL;
I_b_myFirstProfibus_Input2 AT %I*:BOOL;
...
I_b_myFirstProfibus_Input127 AT %I*:BOOL;

Q_b_myFirstProfibus_Output0 AT %Q*:BOOL;
Q_b_myFirstProfibus_Output1 AT %Q*:BOOL;
Q_b_myFirstProfibus_Output2 AT %Q*:BOOL;
...
Q_b_myFirstProfibus_Output127 AT %Q*:BOOL;

END_VAR

```

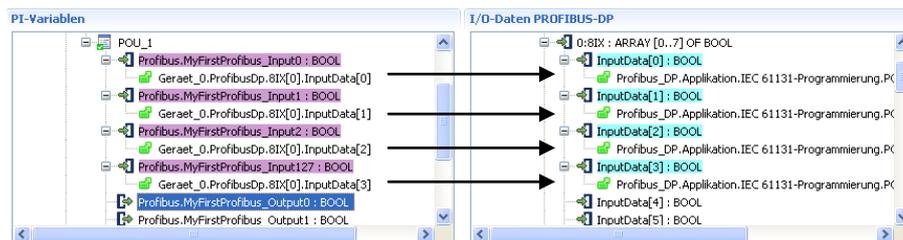
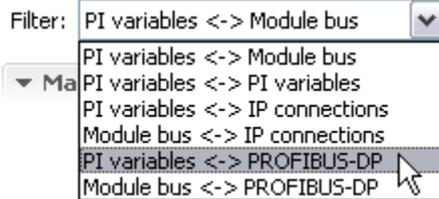




- 1 [Project Management] ペインでフィルタとして選択して、Profibus-DP Slave Editor を開きます。

## I/O mappings: Training\_Sys

Filter for the I/O mappings



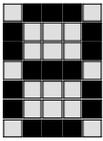
### 3.3 GSDファイル

データ交換が機能するためには、適切な仮想 PROFIBUS モジュールを PROFIBUS-DP マスタと PSSu システムの両方にコンフィグレーションする必要があります。送信データ用の仮想 PROFIBUS モジュールは、出力モジュールとして示されます。

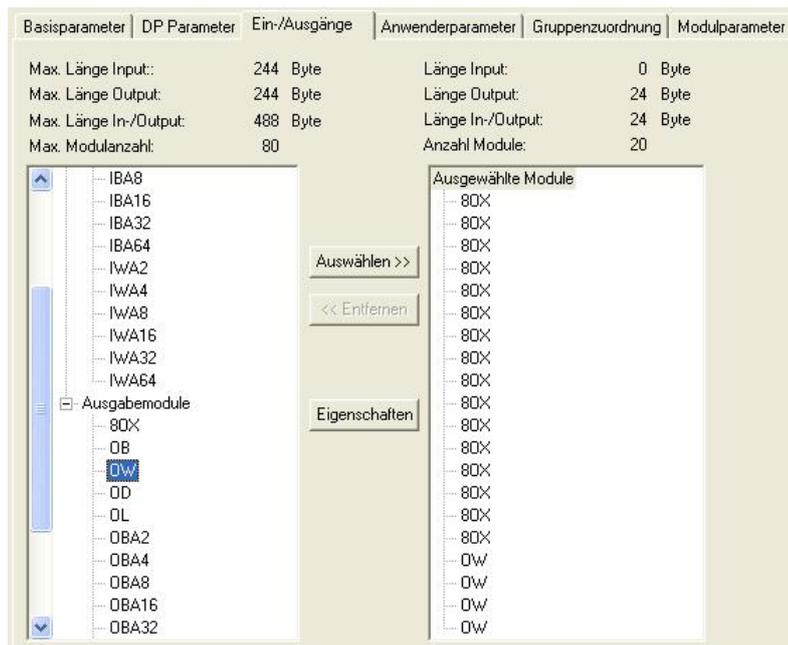
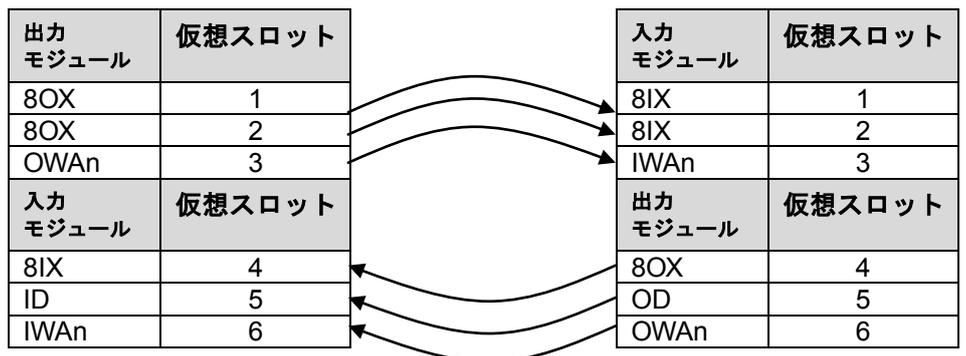
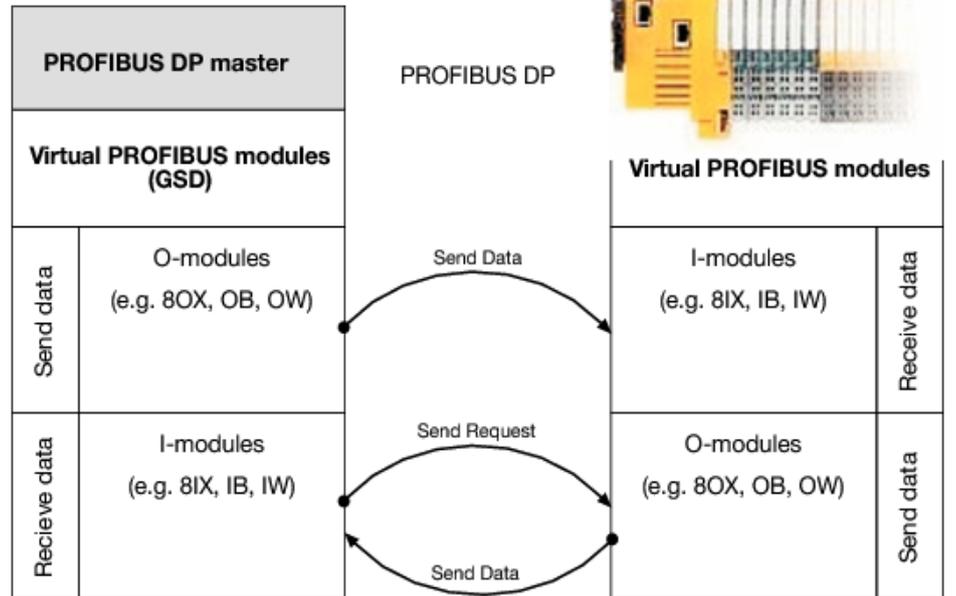
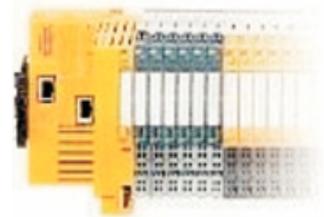
PROFIBUS-DP マスタは必要なデータを、仮想 PROFIBUS モジュールから GSD ファイルで受信します。ファイルは PAS4000 で提供され、ピルツのホームページの [Downloads] セクション ([www.pilz.de](http://www.pilz.de) -> Downloads で「GSD」を検索) から入手することもできます。



登録ユーザのみがピルツのWebサイトからソフトウェアをダウンロードできます。登録は無料です。



PSSu system



第 8 章の終わり

