PVIS (Pilz Visualization)



		目次
1	診断	コンセプトPVIS2
	1.1	PSS 40003
	1.1.1	手順
	1.1.2	基本診断項目の編集7
	1.2	PC13
2	Mod	busTCP接続: PSS 4000 <-> PMI-Pro15
	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*)
	2.1.3	PI変数 (AT) のマッピング18
	2.2	PMI-Proでの設定:19
3	Profi	ibus-DP経由でのフィールドバス接続22
	3.1	要件22
	3.2	Profibus-DP Slave Editor
	3.2.1	Profibus-DPの設定例24
	3.2.2	プログラム内のPI変数 (AT %l*およびAT %Q*)
	3.2.3	PI変数 (AT) のマッピング26
	3.3	GSDファイル27

pilz

2011 年 12 月





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。

- 「 PMIvisu (フルグラフィックスタッチスクリーンディスプレイ)
- ^(愛) 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。

- 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。

Z





- 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。

 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。



- 🗠 Add New Device Select the product type. PSSu PLC Filter: ¥ PSSu PLC Product type su ije Firmware ve Order number: 312070 < Back Next > Finish Cancel
- ❷ 製品タイプを選択します。この場合は [PSS 4000-PC] です。

- [Device Name] はデフォルトの名前 (例: 「NB-KDELEON」) をそのま ま使用します。

DIZ





Add New Device Enter a device name, Device name NB-KDELEON	
Enter a device name. Device name NB-KDELEON	
Device name NB-KDELEON	
Device name NB-KDELEON	
IP address: 192 . 168 . 1 . 2	
< Back Next > Fin	ish Cancel

[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。 Enish 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。

Hardware configuration
 Geraet_0
 Geraet_1
 Geraet_1
 MR-KDELEON

(P

(P

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。

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



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)]

🔥 Configure OPC Server	<u>a</u> 🛛
Configure OPC Server	
Select the device containing th	e OPC Server.
Device with OPC Server:	NB-KDELEON (192,168,1.2)
?	OK Cancel

③ OK をクリックして確定します。

CP 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。

PVIS (Pilz Visualization)



1.1.2 基本診断項目の編集

Ē

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。

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



ΪZ







 ・ 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。

rioriti	es		
dit the	priorities.		
	Priority		
1	<click here="" th="" to<=""><th>edit></th><th></th></click>	edit>	
2	<click here="" td="" to<=""><td>edit></td><td></td></click>	edit>	
3	Error		
4	<click here="" td="" to<=""><td>edit></td><td></td></click>	edit>	
	<click here="" td="" to<=""><td>edit></td><td></td></click>	edit>	
7	<click here="" td="" to<=""><td>edit></td><td></td></click>	edit>	
1	Warning	ourse	
	<click here="" td="" to<=""><td>edit></td><td></td></click>	edit>	
0	<click here="" td="" to<=""><td>edit></td><td></td></click>	edit>	
1	<click here="" td="" to<=""><td>edit></td><td></td></click>	edit>	
2	<click here="" td="" to<=""><td>edit></td><td></td></click>	edit>	
.3	Status informa	tion - Jies	
9 5	Click here to	eait>	
J	SCIICK HERE CO	our.×	
?			OK Cancel
? 象は:	コンボボッ	クスで選択できます	OK Cancel
⑦ 象は: <u>cope</u> ;	コンボボッ	クスで選択できます Operation	OK Cancel
? 象は: <u>cope:</u>	コンボボッ e text:	クスで選択できます Operation	OK Cancel
⑦ 柔(よ: <u>cope:</u> essag	コンボボッ e text:	クスで選択できます Operation Hardware	OK Cancel
⑦ 柔は: :ope: essag	コンボボッ e text:	クスで選択できます Operation Hardware Project configuration	OK Cancel
? ≹(よ: :ope: essag	コンボボッ e text:	クスで選択できます Operation Hardware Project configuration Operation	OK Cancel
? 良は: :ope: essag	コンボボッ e text:	クスで選択できます Operation Hardware Project configuration Operation Plant	OK Cancel
? 象は: <u>cope:</u>	コンボボッ e text: コン:	クスで選択できます Operation Hardware Project configuration Operation Plant ボボックス内のテ ⁴	ок Cancel
⑦ 象はこ cope:	コンボボッ e text: コン: 「500	クスで選択できます Operation Hardware Project configuration Operation Plant ボボックス内のテ ^ュ 20:11 で変更するこ	ок Cancel t。 Fストは、ハイパー とができます。
② 東は: cope:	コンボボッ e text: コン: 「500	クスで選択できます Operation Hardware Project configuration Operation Plant ボボックス内のテ ^ュ De: 」で変更するこ	ок Cancel す。 トストは、ハイパー とができます。
② 象は: cope: essag	コンボボッ e text: コン: 「500	クスで選択できます Operation Hardware Project configuration Operation Plant ボボックス内のテ ⁴ Det	ок Cancel t。 Fストは、ハイパー とができます。
② 象は: cope: essag	コンボボッ e text: コン: 「500	クスで選択できます Operation Hardware Project configuration Operation Plant ボボックス内のテキ	ок Cancel t。
② 家は: cope: copes	コンボボッ e text: コン: 「500	クスで選択できます Operation Hardware Project configuration Operation Plant ボボックス内のテ ⁴ 20:1 」で変更するこ	ок Cancel t。 Fストは、ハイパー とができます。

Ē

Ø

Ē

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。

Message text:

Hardware Project configuration Operation Plant



ilz





▼ Info	rmation				
Date cr	reated :	2011:06:17 13:	40:13		
Last mo	odified :	2011:06:17 13:	40:13		
Author	:	Pilz GmbH & Co.	KG		
Comme	ent :			~	
• [Cor	ndition]				
Conditio	DN				
Conditio	n for activat	ing a diagnostic m	essage		
Kind :	Value	~			
Variable			~		
Delay ti	me: 00:00: [Conc	00 🛃 lition] は、変数	なを指定して直接	をBDI を割り付ける	っま
Delay ti を数のコ existing	me: 00:00: [Conc 可能(: ンテキス BDI] を選	00 粂 lition]は、変数 こなりません。 - メニューから 択します。	なを指定して直接 [Process Diag	度 BDI を割り付ける J nostic] → [New B	っ す DI
Delay ti を数のコ existing ∋ prog	me: 00:00: [Conc 可能に ンテキスト BDI] を選 RAM Tra:	00 lition]は、変数 こなりません。 、メニューから 択します。 ining_Syste	なを指定して直接 【 Process Diag m	度 BDI を割り付ける J nostic] → [New B	っま DI
Delay ti 変数のコ existing ⊖ PROGI ⊖ VAR	me: 00:00: [Conc 可能に ンテキスト BDI] を選 RAM Tra:	00 lition] は、変数 こなりません。 、メニューから 択します。 ining_Syste	なを指定して直接 [Process Diag m	度 BDI を割り付ける J nostic] → [New B	っま DI
Delay ti 変数の⊐ existing ● PROGI ● VAR	me: 00:00: [Conc 可能に ンテキス BDI] を選 RAM Tra: outg 少い	00 ◆ lition]は、変数 こなりません。 、メニューから 択します。 ining_Syste	なを指定して直接 [Process Diag m 	度 BDI を割り付ける Jnostic] → [New B	っま DI
Delay ti 変数のコ existing PROGI VAR	me: 00:00: 「Conc 可能に ンテキスト BDI]を選 RAM Tra: outr グロ CAL	00 lition] は、変数 こなりません。 、メニューから 択します。 ining_Syste	なを指定して直接 「Process Diag em Ctrl+Z Ctrl+X Ctrl+X	g BDI を割り付ける J nostic] → [New B	っま DI
Delay ti 変数の⊐ existing ● PROG ● VAR	me: 00:00: [Conc 可能に ンテキス日 BDI]を選 RAM Tra: out; グロ CAL 00) 00	00 lition] は、変数 こなりません。 、メニューから 択します。 ining_Syste	なを指定して直接 [Process Diag em Ctrl+Z Ctrl+Z Ctrl+Y Ctrl+Y	度 BDI を割り付ける jnostic] → [New B	っま DI
Delay ti 変数のコ existing PROGI VAR	me: 00:00: [Conc 可能に ンテキスト BDI]を選 RAM Tra: out; グロ CAL CO CAL CO) CAL Pa	00 lition] は、変数 こなりません。 、メニューから 訳します。 lining_System ndo ut ppy astemptic former in	なを指定して直接 「Process Diag em Ctrl+Z Ctrl+Z Ctrl+X Ctrl+Y	g BDI を割り付ける Jnostic] → [New B	っま DI
Delay ti 変数の⊐ existing ● PROGI ● VAR	me: 00:00: [Cond 可能(: ンテキス BDI]を選 RAM Tra: out; ジロ CAL 00) CAL 00) CAL 00) CAL 00 0 CAL 00	00 lition] は、変数 こなりません。 、メニューから 択します。 ining_Syste ndo ut ppy aste	なを指定して直接 「Process Diag em Ctrl+Z Ctrl+Z Ctrl+Y Ctrl+Y Ctrl+V F9	度 BDI を割り付ける Jnostic] → [New B	っま DI
Delay ti 変数のコ existing	me: 00:00: 「Conc 可能!: ンテキス! BDI]を選 RAM Tra: out; ジロ CAL CO CAL CO CAL CO CAL AC Swit Fo	00 全 lition] は、変数 こなりません。 、メニューから 択します。 ining_Systendo ut oppy aste	なを指定して直接 「Process Diag em Ctrl+Z Ctrl+Z Ctrl+K Ctrl+C Ctrl+V F9 Ctrl+Shift+F	g BDI を割り付ける Jnostic] → [New B	っま DI
Delay ti 変数の⊐ existing ♥ ROGI ♥ VAR	me: 00:00: 「Conc 可能に ンテキスト BDI]を選 RAM Tra: outr ジロ CAL CC) CAL CC) CAL CC) CAL Fr Auto Pr	00 lition] は、変数 こなりません。 、メニューから 択します。 ining_Syste ining_Syste oggle Comment dd from library ormat ocess Diagnostics	なを指定して直接 「Process Diag em Ctrl+Z Ctrl+Z Ctrl+X Ctrl+C Ctrl+V F9 Ctrl+Shift+F	度 BDI を割り付ける jnostic] → [New B) DI
Delay ti 変数のコ existing VAR	me: 00:00: 「Conc 可能! ンテキスト BDI]を選 RAM Tra: Out; ジロ CAL CO CAL CO CO CAL CO CAL CO CO CAL CO CO CAL CO CO CAL CO CO CAL CO CO CAL CO CO CAL CO CO CAL CO CO CO CAL CO CO CO CO CO CO CO CO CO CO	00 全 lition] は、変数 こなりません。 、メニューから 択します。 ining_Syste ndo ut oppy aste oggle Comment dd from library ormat occess Diagnostics rnamic program disc	なを指定して直接 [Process Diag em Ctrl+Z Ctrl+Z Ctrl+X Ctrl+V F9 Ctrl+Shift+F	度 BDI を割り付ける pnostic] → [New B New BDI: From New New BDI: Use existing	5 J BD



8

(F



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。

reate basic diagnostic item	
Select the basic diagnostic item that will	I serve as the copy template for the new basic diagnostic item.
Basic diagnostic item	
PS_EmergencyStop.BDI_Opera	vForReset
<pre>% FS_EmergencyStop.BDI_Read</pre>	yForTest
- 2 FS_EmergencyStop.BDI_Switch	hError
- PS_EmergencyStop.BDI_Input	NotValid
BDI_Operated_1	
<	
Name of the basic diagnostic item	
C - 1 🔽 - 1	
?	Finish Cancel
Beenden を押して手順	を完了します。
Beenden を押して手順	を完了します。 _{変化・} var1: BOOL; _→ var1: BOOI
<u>Beenden</u> を押して手順 1. ステータスの	を完了します。 変化: <mark>var1: BOOL; _う <u>var1</u>: BOOI</mark>
^{Beenden} を押して手順 1. ステータスの	ēを完了します。 変化: <mark>var1: BOOL; _→ <i>var1</i>: BOOI</mark>
^{Beenden} を押して手順 1. ステータスの	ēを完了します。 変化: <mark>var1: BOOL;</mark> → <mark>var1</mark> : BOOI
Beenden を押して手順 1. ステータスの	ēを完了します。 変化: <mark>var1: BOOL;</mark> → <mark>vær1</mark> : BOOI
^{Beenden} を押して手順 1. ステータスの [Remedy]	ēを完了します。 変化: <mark>var1: BOOL;</mark> → <mark>vær1</mark> : BOOI
Beenden を押して手順 1. ステータスの [Remedy]	ēを完了します。 変化: <mark>var1: BOOL; →</mark> <u>var1</u> : BOOI
Beenden を押して手順 1. ステータスの [Remedy] ^{デ New Action} ボタンを	ēを完了します。 変化: <mark>var1: BOOL; →</mark> <u>var1</u> : BOOI 使用して他のアクションをリストに追加し
Beenden を押して手順 1.ステータスの [Remedy] ^{ア New Action} ボタンを Delete ボタンを使月	ēを完了します。 変化: <mark>▼ar1: BOOL; →</mark> <u>vær1</u> : BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま
Beenden を押して手順 1.ステータスの [Remedy] ^{テ New Action} ボタンを Delete ボタンを使月	ēを完了します。 変化: <mark>▼ar1: BOOL; → <mark>▼ær1</mark>: BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま</mark>
Beenden を押して手順 1.ステータスの [Remedy] ^{テ New Action} ボタンを Delete ボタンを使月	iを完了します。 変化: <mark>var1: BOOL;</mark> → <u>var1</u> : BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま
Beenden を押して手順 1.ステータスの [Remedy] ^{ア New Action} ボタンを Delete ボタンを使月	を完了します。 変化: var1: BOOL; → var1: BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま ies ⊠
Beenden を押して手順 1. ステータスの [Remedy] ^{ア New Action} ボタンを Delete ボタンを使月	iを完了します。 変化: var1: BOOL; → var1: BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま ies &
Beenden を押して手順 1.ステータスの [Remedy] ^{デ New Action} ボタンを Delete ボタンを使月 ^C Problems Propert ² BDI Type: Training	を完了します。 変化: var1: BOOL; → <u>vær1</u> : BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま ies & 3_System.BDI_1
Beenden を押して手順 1. ステータスの [Remedy] デ New Action ボタンを Delete ボタンを使用 Problems Propert 2 BDI Type: Training	を完了します。 変化: var1: BOOL; → var1: BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま ies ☆ 3_System.BDI_1
Beenden を押して手順 1. ステータスの [Remedy] ^{デ New Action} ボタンを Delete ボタンを使月 Celete ボタンを使月 Problems Propert 2 BDI Type: Training General	を完了します。 変化: var1: BOOL; → <u>vær1</u> : BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま ies Ø <u>3_System.BDI_1</u> Remedy
Beenden を押して手順 1. ステータスの [Remedy] 『 New Action ボタンを Delete ボタンを使用 Problems Propert ジ BDI Type: Training General Remedy	を完了します。 変化: var1: BOOL; → vær1: BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま ies Ø 3_System.BDI_1 Remedy
Beenden を押して手順 1. ステータスの [Remedy] 『New Action ボタンを Delete ボタンを使用 Problems Propert Problems Propert PBDI Type: Training General Remedy	を完了します。 変化: var1: BOOL; → <u>vær1</u> : BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま ies 窓 3_System.BDI_1 Remedy New Action Delete
Beenden を押して手順 1. ステータスの [Remedy] ^⑦ New Action ボタンを Delete ボタンを使用 ? Problems Propert ? BDI Type: Training General Remedy Location Information	iを完了します。 変化: var1: BOOL; → var1: BOOI 使用して他のアクションをリストに追加し 目してアクションをリストから削除できま ies Ø g_System.BDI_1 Remedy New Action Delete
Beenden を押して手順 1. ステータスの [Remedy] 『New Action ボタンを Delete ボタンを使用 Problems Propert Problems Propert Poblems Remedy General Remedy Location Information	を完了します。 変化: var1: BOOL; → vær1: BOOI 使用して他のアクションをリストに追加し 用してアクションをリストから削除できま ies 窓 J_System.BDI_1 Remedy New Action Delete Remedy
Beenden を押して手順 1. ステータスの [Remedy] New Action ボタンを Delete ボタンを使用 Problems Propert Poblems Propert Ceneral Remedy Jocation Information	iを完了します。 変化: var1: BOOL; → var1: BOOI 使用して他のアクションをリストに追加し 目してアクションをリストから削除できま ies Ø g_System.BDI_1 Remedy New Action Delete Remedy



2	

Action text:			
Scope :		~	
Responsibility :		~	
Location info	mation		
 Pind: ① [Action Text] フィ ② 対象はコンボボ ③ 担当者はコンボ ④ 位置情報はコン b. [Location i 	Sancor/actuator ィールドにテキス ックスで選択し、 ボックスで選択し ボボックスで選択 nformation] 置情報の種類は、	トを入力します。 ハイパーリンクを使っ 、ハイパーリンクを使っ できます。 コンボボックスで指定	って編集できます。 もって編集できます。 ■できます。
🖹 Problems 🗍	Properties 🛛		
𝒫 BDI Type:	Training_System.E	3DI_1	
General			
Remedy	Kind:	Sensor/actuator	
	nation	Terminal Sensor/actuator POU User-defined	







ΪZ





🥖 C: Wokumente u	n d Einstellu Dokumente und	ngen\kdeleon\L d Einstellungen\kdel	.okale Einstellu eon\Lokale Einstel	ungen\Temp\Temp lungen\Temp\Temporär	oräres Ver res 🕶 🔸	zeichnis 3 für P - Wind × Google	¥□ا <u>گ</u> يا - ۹
<u>D</u> atei <u>B</u> earbeiten <u>A</u>	Insicht <u>F</u> avo	riten E <u>x</u> tras <u>?</u>					
🚖 🏟 🌈 C:\Doku	imente und Ein	stellungen\kdeleon\	Lokale Einst		6	i • 🖸 • 🖶 • 🕞 s	eįte 🔹 🌍 Extras 🔹 🎽
😣 i	Projekt: Diagnos	Training_Sys seliste (1 / 4)	tem_Service	•			
		t		Info (F2)		Abhilfe (ENT)	Param. (ESC)
C Error DID: Bereich: Gerät: BMK: Ortsbeschr.: Adresse: Beschreibung:	17.06.2011 P-0001-000 Operation E-STOP p	13:56:55 D1 Jshbutton has br	een operated.				Ш
1 Status infor DID: Bereich: Gerät: BMK: Ortsbeschr.: Adresse: Beschreibung:	17.06.2011 S-0303-000 Project con Device_1 Project is p	13:36:25 D1 nfiguration unlicensed.					
1 Status infor DID: Bereich: Gerät: BMK: Ortsbeschr.: Adresse: Beschreibung:	17.06.2011 S-0303-000 Project con Device_0 Project is	13:36:16 D1 nfiguration unlicensed.					
Fertig					🗟 Eiaer	ner Computer	₹ 100% -



2 ModbusTCP接続: PSS 4000 <-> PMI-Pro

2.1 PAS4000 での設定

Ē

● 次の設定で、新しい IP 接続をセットアップします。

 → ↓ ↓ → ↓ ↓ ↓	onfiguration _0 resource resource vice_0 connections	
🕀 💐 Device	Open IP Conne	ctions Editor
🖽 👷 NB-KDI	Сору	Ctrl+C
	💼 Paste	Ctrl+V
	🗙 Delete	Del
	Rename	F2
	Compare	
	Properties	Ctrl+Enter
L		

IP connections edi	tor (Device_0) 🛛					
IP Connec	tions (Device: De	evice_0)				
Configured Conn	ections					
Duchasel	Constanting Marrie	Dela	Develo TD Address	Desighe Death Marsha		
Protocol	Connection Name	RUIE	Remote IP Address	Remote Port Numbe	New Modbus/TCP Server	
					New Modbus/TCP Client	
					New Raw UDP Connection	
<				>	Delete	~
Ionnections Serve	er mapping table for Modbus	/TCP				

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。

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。

🚇 *IP connections editor (Device_0) 🛛 🖓 🗖							
🚇 IP Connecti	ons (Device: De	vice_0)					
Configured Connec	tions						
Protocol	Connection Name	Role	Remote IP Address	Remote Port Number	n		
Modbus/TCP	ModbusTCPServer_0	Server	0.0.0.0	0			
					New Modbus/TCP Server		
					New Modbus/TCP Client		
					New Raw UDP Connection		
					Delete		
Network settings					^		
Connection Name	ModbusTCPServer_0	Keep alive	settings eep alive				
Unit ID	255	Keep alive ti	me [ms] 7200000		=		
Local port number	502	Keep alive in	nterval [ms] 1000				
Remote IP Address	0.0.0.0	Connection timeout					
Remote Port Number	0	Enable Connection Timeout					
		Connection	cycles 10				
		Timeout = c	onnection cycles x connect	tion cycle time	~		
Connections Server	manning table for Modbus(1	CP.	ш		>		
a conneccond b boryor	mapping cases for modelasy i	-					

ilz



④ 接続名と IP アドレスを入力します。

^{CP} 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。

- > $0xRead \rightarrow BOOL(コイル)$
- ➢ 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)

🛃 *IP connections edit	tor (Device_0) 🕅			- 6
E Server map	ping table	for Modbus	F/TCP (Device: Device_0)	
Data Range	Start Address	Data Length	Data update cycle time	
0xRead	0	10	Calculate automatically	
0×Write	0	10	Data update cycle time [ms]	
1xRead	0	0		
3xRead	0	0	100	
4xRead	0	10		
4×Write	0	10		
Data range settir	ng			
Start address 0				
Data length 10				
Connections Server	mapping table for	Modbus/TCP		

Ē

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。

<u>1. PSS 4000 がサーバの場合 (推奨設定)</u>

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%l*

Z



<u>2. PSS 4000 がクライアントの場合</u>

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%l*

Ē

クライアントによって読み取られた入力変数 (%1*)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	
<pre>I_b_input_safebool AT %I*:SAFEBOOL; Q_b_output_safebool AT %Q*:SAFEBOOL; Q_b_output_safebool_2 AT %Q*:SAFEBOOL;</pre>	//Hardware Input 1 //Hardware Output 1 //Hardware Output 2
<pre>I_w_input_data_word AT %I*:WORD; Q_w_output_data_word AT %Q*:WORD;</pre>	//WORD Variable written by PMI-Pro //WORD Variable read by PMI-Pro
<pre>I_b_input_data_bool AT %I*:BOOL; Q_b_output_data_bool AT %Q*:BOOL;</pre>	//BOOL Variable written by PMI-Pro //BOOL Variable read by PMI-Pro
<pre>I_b_input_data_bool_2 AT %I*:BOOL; Q_b_output_data_bool_2 AT %Q*:BOOL;</pre>	//BOOL Variable written by PMI-Pro //BOOL Variable read by PMI-Pro
<pre>Q_b_output_data_bool_3 AT %Q*:BOOL;</pre>	//BOOL Variable read by PMI-Pro
END_VAR	
(*PMI-Pro able to change WORD-Variab	les*)
LD I_w_input_data_word	
ST Q_w_output_data_word	
(*PMI-Pro able to change BOOL-Varia	ables *)
LD I b input data bool	
ST Q_b_output_data_bool	
(*PMI-Pro able to set Hardware-Out] LD I_b_input_data_bool_2	put 1*)
EQ TRUE	
ST Q_b_output_safebool	
ST Q_b_output_data_bool_2	<pre>// PMI-Pro read Status of Hardware-Output 1</pre>
(*PMI-Pro able to set Hardware-Out	put 1*)
LD I_b_input_safebool	-
EQ TRUE	
ST 0_b_output_safebool_2	
ST 0_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:

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

efinition of driver		
Driver name:		
MODBUS RTU MASTER		<u>o</u> k
Available drivers:		<u> </u>
	_	Help
🗄 🛅 CANopen		
🗄 💼 HITACHI		
🗄 👝 INTERBUS		
🗄 👝 MITSUBISHI		
🗄 🔄 MODBUS		
± OMRON		
🗄 💼 OPC		
🗄 💼 PILZ	-	
÷ 👝 phoripue		
Priver information:		
Modbus RTU Master		
Operating systems: WIN98, WinNT4.0, Win2000, WinCE		
Version: V5.5.0.20 Cevial and TCD //D		
Senai and TUP7IP		
		V

❷ [Option] タブで TCP/IP 接続ドライバを設定します。

eneral Com Option Com. TCP.	1P	
Connection: O Serial	C ICE/IB	<u> </u>
Oider for FLOAT objects O HB Controler FLOAT O Standard Modbus FLOAT	Order for DWORD objects: Intel Format DWORD Motorola Format DWORD	
FTS Handshake Offset 1	TCP - configuration File: Default.iso	
max. Block size: 32	Send Delay:	
Reconnect Timeout:		



Ē

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。 ● [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 Configuration × General Com Option Conn. TCP/IP <u>0</u>K Default.iso Conn. File: <u>C</u>ancel Connections: Connectionlist: 1 Busaddress: Verbindungsnamen IPSS4000 HW PSS4000 Connectionname: Remote IP Address: 192.168.1.175 Remote Portaddress: 502 <u>N</u>ew <u>D</u>elete <u>E</u>dit <u>P</u>ing 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。 × Configuration project Runtime Directories Colour Remote transport RT changeable data <u>0</u>K CE - Project basic path: Transport device: Configuration parameters: TCP/IP HOST=192.168.1.94; <u>C</u>ancel active Destination Revision Description 🔺 source <u>H</u>elp 2 E:\Customer_Support\FC:\IPSM\Data Сору Basic project patł X 3 \Bitmaps Сору Bitmap path (*.br X 4 **\Lists** Сору List path (*.txt,*.q

Сору

Сору

Сору

Copy

C:\IPSM\Data

Video path (*.avi)

Report path (*.xrs

pilz

5 🕱 Wideo

6 🕱 \Tables

7 🗙

8 🗖

\Default.iso





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。

Select va	riable									1
Search	Data_Bool		Variables - in total:4	Licensed I/O's	unlimited					Exit
		Nama	selected: 4	Used I/U :	ena Add	vaaa Mada		Driver	Beedien metrix	
Data	Bool	Name		TAG-Nr	C (1) 000	0.00 V	NO	DRTU32/MODBUS RTU MASTER	Reaction matrix	Help
Data	Vord			TAG-Nr TAG-Nr	HR (1) 00	00.00 V	NO	DRTU32/MODBUS RTU MASTER		
Outp	1t_1 1t_2			TAG-Nr TAG-Nr	C (1) 000	2.00 ¥	HO	DRTU32/MODBUS RTU MASTER		
										<u>N</u> ew
										Change
										Delete
										Сдру
										Eilter







3 <u>Profibus-DP経由でのフィールドバス接続</u>

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。

Filter:	[PSSu PLC	~]	
Product typ Firmware ve Order numb	e: ersion: er:	PSSu H PLC1 FS DP PSSu H PLC1 FS SN : PSSu H PLC1 FS SN : PSSu H PLC1 FS DP : 312071	SN SD 🔽]	
X3 Grebwer p X4			MBUS SD-Card ST Run DKAS ST FORCE SF BF ST SNp FS SNp SD	SV 24V X3: LNK X3: TRF	
PSS 4000	PROFIB	US DP PSSu H PLCIFS DP SNSC		24V 24V OV OV X1 24V C OV (\$\.).22	

3.2 Profibus-DP Slave Editor

 [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。

8-22

7





Device_2:PR	OFIBUS-DP Slave	Editor 🕄	
PROFIBUS	5-DP Slave	Editor	^
Selection of t Select the mod I-modules IB IB ID IW IWAn	he virtual PRC lule type, the nu O-modules 80X 0B 0BAN 0D 0W 0W 0WAN	FIBUS modules nber of modules of this module type and the number of elements (if required) that you wish to add. Number of bytes required: 1 Number of modules: 1 Number of elements: 2 Add	H
Configured vi	irtual PROFIBU	5 modules Other Settings	
Module	virtual slot	Number or bytes and modules stul available White black that has dagheded Receive data [Byte]: 244 Send data [Byte]: 244 Modulast Bn	item

② 次の I/O モジュールを実装できます。

I-modules	入力 モジュール	意味	変数の宣言
IBAn	8IX	バイナリ入力 x 8	AT %I* : BOOL (8 回)
ID	IB	バイト入力 x 1	AT %I* : BYTE
IW	IW	ワード入力 x 1	AT %I* : WORD
IWAN	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 [063] OF BYTE
	IWAn	入力配列 (ワード形式) x 1	AT %I* : ARRAY [063] OF WORD

Ē

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 80X	出力 モジュール	意味	変数の宣言
OB	8OX	バイナリ出力 x 8	AT %Q* : BOOL (8 回)
OD	OB	バイト出力 x 1	AT %Q* : BYTE
ow	WO	ワード出力 x 1	AT %Q* : WORD
OWAn	OD	ダブルワード出力	AT %Q* : DWORD
		x 1	
	OBAn	出力配列 (バイト形	AT %Q* : ARRAY [063] OF
		式) x 1	BYTE
	OWAn	出力配列 (ワード形	AT %Q* : ARRAY [063] OF
		式) x 1	WORD

Ē

出カ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。

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	入力	出力	モジュール数	ファンクション数
Number of modules: 16	モジュール	モジュール		
Number of elements	8IX	8OX	1 … 80	
	IB	OB	1 … 80	
Add			•	

ilz





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

[●] 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	モジュール数/			
		モジュール	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。						
送信デー	-タ長 = 32 バイト。	内訳は以下の通	19.			
送信デー バイト	-タ長 = 32 バイト。 変数のデータ型	内訳は以下の通 入力	iり。 モジュール数/			
送信デー バイト	- 夕長 = 32 バイト。 変数のデータ型	内訳は以下の通 入力 モジュール	iり。 モジュール数/ ファンクション数			
送信デ- バイト 16	-タ長 = 32 バイト。 変数のデータ型 BOOL	内訳は以下の通 入力 モジュール 8IX	iり。 モジュール数/ ファンクション数 16/-			

入力モジュール [8|X] を選択して、モジュール数を 16 に設定します。 Add をクリックして確定します。

I-modules O-modules

80X	Number of bytes requ	ired:	16
OB	Number of modules:	16	
OBAn			N
OD	Number of elements:	2	~
OW			
OWAn	Add		
	80X OB OBAn OD OW OWAn	80X Number of bytes required 0B Number of modules: 0BAn Number of elements: 0D Number of elements: 0W Add	80X Number of bytes required: 0B Number of modules: 16 0BAn Number of elements: 2 0D Number of elements: 2 0W Add

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	
	Delete	





 入力モジュール [IWAn] 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	0-modules	
8IX	80X	Number of bytes required: 8
IB	OB	Number of modules: 1 🚔
IBAn	OBAn	
IW	OU OW	Number of elements: 4
IWAn	OWAn	Add 2 1
010	14	
8IX	15	
IWA4	16	~

 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 [80X] と [OWAn])。

Configured virtual PROFIBUS modules

Module	Virtual slot	^
80X	22	
80X	23	
80X	24	
80X	25	
80X	26	
80X	27	
80X	28	
80X	29	
80X	30	_
80X	31	=
80X	32	
OWA4	33	-
	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_Output2 AT %Q*:BOOL; ... END_VAR

ĬZ



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)。







● [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/	/0 n	nappings: Trainir	ng_S	ys [.]
F	Filter fo	or the I/O mappings		
F	Filter:	PI variables <-> Module bu	s [~
•	▼ Ma	PI variables <-> Module bu PI variables <-> PI variable PI variables <-> IP connect Module bus <-> IP connect PI variables <-> PROFIBUS Module bus <-> PROFIBUS	s ions ions -DP	
PI-Varial	blen			I/O-Daten PROFIBUS-DP
4		Porfbus.MyFirstProfbus_Input0 : BOOL aff Geraet_0.ProfibusDp.81(0].Input0 ato[0] Profbus.MyFirstProfbus_Input1 : BOOL aff Geraet_0.ProfibusDp.81X(0].Input0 ato[1] eff Geraet_0.ProfibusDp.81X(0].Input0 ato[2] aff Geraet_0.ProfibusDp.81X(0].Input0 ato[2] aff Geraet_0.ProfibusDp.81X(0).Input0 ato[2] aff Geraet_0.ProfibusDp.81X(0).Input0 ato[2] aff Geraet_0.ProfibusDp.81X(0).Input0 ato[2] aff Geraet_0.ProfibusDp.81X(0).Input0 ato[3] aff Geraet_0.ProfibusDp.81X(0).Input0 ato[3] Profibus.MyFirstProfibus_0.000004 Profibus.MyFirstProfibus_0.00001		OBIX: ARRAY [07] OF BOOL OBIX: ARRAY [07] OF BOOL

3.3 GSDファイル

Ē

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 -> 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。









Basisparameter DP Parameter	Ein-/	Ausgänge	Anwenderparameter Gruppe	nzuordnun	ig Mod	Julparameter
Max. Länge Input::	244	Byte	Länge Input:	0	Byte	
Max. Länge Output:	244	Byte	Länge Output:	24	Byte	
Max. Länge In-/Output:	488	Byte	Länge In-/Output:	24	Byte	
Max. Modulanzahl:	80		Anzahl Module:	20		
BA8 		-	Ausgewählte Module 80X			
IBA32			80×			
- IBA64						
IWA2		Auswa	hlen >> 80×			
IWA4		and the second second	80×			
IWA8		<< Er	tfernen 80X			
IWA16			80X			
IWA32			80×			
IWA64			80×			
Ausgabemodule		Eigens	chaften 80X			
- OB			80×			
- NW			80×			
			80×			
01						
- OBA2						
- OBA4			- ow			
- OBA8			- ow			
OBA16			- ow			
- OBA32			- ow			

第8章の終わり