

SIEMENS S7 シリーズ用 S7NetPlus プロバイダ

Version 1.0.1

ユーザーズ ガイド

October 26, 2020

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2019-07-11	初版.
1.0.1	2019-09-17	DataBlock 以外へのアクセスに対応
	2020-10-26	付録 A の誤記修正

【動作確認機種】

機種		注意事項
S7-1500		機能に一部制限があります.
S7-300		

目次

1. はじめに	5
2. プロバイダの概要	6
2.1. インストール	6
2.2. 概要	6
2.3. 制約	6
2.4. メソッド・プロパティ	7
2.4.1. CaoWorkspace::AddController メソッド	7
2.4.2. CaoController::AddVariable メソッド	8
2.4.2.1. DBNo オプション	10
2.4.3. CaoVariable::get_Value メソッド	11
2.4.4. CaoVariable::put_Value メソッド	11
3. Variable の Type の詳細	12
3.1. Bool	12
3.1.1. Address	12
3.1.2. Elem	12
3.1.3. 注意	12
3.2. Char, Byte, Int, Word, DInt, DWord, LInt, LWord, Real, LReal	12
3.3. String, Wstring	13
3.4. CharArray, WCharArray	13
3.4.1. 読み込み	13
3.4.2. 書き込み	13
4. 機種ごとの制約事項	14
4.1. アクセスできる DataType の制約	14
4.2. 指定できる値の型の制約	14
5. ライセンス表記	15
5.1. S7.Net+	15
付録 A. アドレス指定例	16
付録 B. 文字列型へのアクセスについて	18

付録 C. リモートパートナーからのアクセス許可について.....	19
-----------------------------------	----

1. はじめに

SIEMENS S7 シリーズ用プロバイダは, SIEMENS S7 シリーズ(S7-200,300,400,1200,1500)に対してアクセスを行う ORiN2 CAO プロバイダです.

本ドキュメントでは, SIEMENS S7 シリーズ用プロバイダの概要と, 実装されている CAO インタフェース(関数仕様)について説明しています.

2. プロバイダの概要

2.1. インストール

SIEMENS S7 シリーズ用プロバイダモジュールは、下記の DLL で構成されています。インストーラでインストールした場合は、インストール作業は不要です。手動でインストールする場合は、表 2-1 のように実行してください。

表 2-1 SIEMENS S7 シリーズ用プロバイダ

ファイル名	GaoProvSIEMENSS7NetPlus.dll
ProgID	GaoProv.SIEMENS.S7NetPlus
レジストリ登録	RegAsm.bat GaoProvSIEMENSS7NetPlus.dll
レジストリ登録の抹消	UnregAsm.bat GaoProvSIEMENSS7NetPlus.dll

※ RegAsm.bat および UnregAsm.bat は ORiN2 インストールフォルダ下の DotNet¥Bat フォルダにあります。

2.2. 概要

SIEMENS S7 シリーズ用プロバイダは、オープンソースライブラリの S7Net+ を用いて、PLC 内部のデータベースに定義されている変数へのアクセスを行います。

データベースの変数定義は SIEMENS の TIAPortal を使って行ってください。

2.3. 制約

本プロバイダは使用しているライブラリの仕様により以下の制約があります。

- ◆ アクセス可能なデータタイプは機種ごとに制限があります。
- ◆ 最適化されたデータベースにはアクセスできません。

2.4. メソッド・プロパティ

2.4.1. CaoWorkspace::AddController メソッド



AddController (<bstrCtrlName:BSTR>, <bstrProvName:BSTR>, <bstrPcName:BSTR>, [**<bstrOption:BSTR>**])

< bstrCtrlName >	:	[in] コントローラ名
< bstrProvName >	:	[in] プロバイダ名. 固定値 ="CaoProv.SIEMENS.S7NetPlus"
<bstrPcName>	:	[in] プロバイダの実行マシン名
<bstrOption>	:	[in] 以下のオプションをコンマ区切りで指定します.
Conn	:	接続先を指定します. (必須) 例)“Conn=TCP:{接続先の IP アドレス}”
CPUType	:	通信を行う CPU の種類をハイフンなしで指定します. (必須) S7200, S7300, S7400, S71200, S71500 の 5 機種が指定できます. 例)“CPUType =S71500”
Rack	:	接続するラックの番号を指定します. 省略時は 0 として扱われます.
Slot	:	接続するスロットの番号を指定します. 省略時は 0 として扱われます.



```
CCaoController ctrl = engine.Workspaces[0].AddController (
    "Test", "CaoProv.SIEMENS.S7NetPlus",
    null, "Conn=TCP:192.168.0.3,CPUType=S71500");
```

2.4.2. GaoController::AddVariable メソッド

変数へアクセスするための CaoVariable オブジェクトを追加します。



AddVariable (<bstrName:BSTR>, [<bstrOption:BSTR>])

- <bstrName > : [in] 変数の名前を指定します。
“@”始まり以外の任意の文字列を指定してください。
- <bstrOption> : [in] 以下のオプションをコンマ区切りで指定します。
- DataType : アクセスするメモリのタイプを指定します。
 DataBlock, Input, Output, Memory, Counter, Timer が指定可能です。
 省略時は DataBlock が指定されたことになりま
 す。
 詳細は[アクセスできる DataType の制約](#)を参照し
 てください。
- Type : 取得する値の型を指定します。
 Bool, Char, Byte, Int, Word, DInt, DWord, LInt, LWord, Real, LReal, String, WString, CharArray, WCharArray が指定可能です。
 DataType に Counter および Timer 以外が指定された場合は必須です。
 DataType に Counter および Timer が指定された場合は Type オプションで指定された値は無視され、Counter の場合は WORD, Timer の場合は LReal が指定されたことになります。
 各型の詳細は [Variable の Type の詳細](#) を参照してください。
 また、指定できる値の型は機種によって制約があります。
 詳しくは[指定できる値の型の制約](#)を参照してください。
- Address : アクセスするメモリのアドレスを指定します。
 Type に Bool を指定した場合は {アドレス}.{ビットオフセット} の形で指定してください。
 Type に Bool 以外を指定した場合はアドレスだけを指定してください。
- DBNo : アクセスするデータブロックの番号を指定しま

す。
DataType に DataBlock 以外を指定した場合は無視されます。
省略時は 1 として扱われます。

Elem : 取得する要素数を指定します。
0 以上の整数で指定してください。
省略時は 0 として扱われます。

また、Type に String および WString を指定した場合、Elem は無視され、0 を指定したことになります。

0 を指定した場合および、Type に CharArray または WCharArray を指定した場合は指定した型として値を取得します。

それ以外の場合は指定した型の配列として値を取得します。

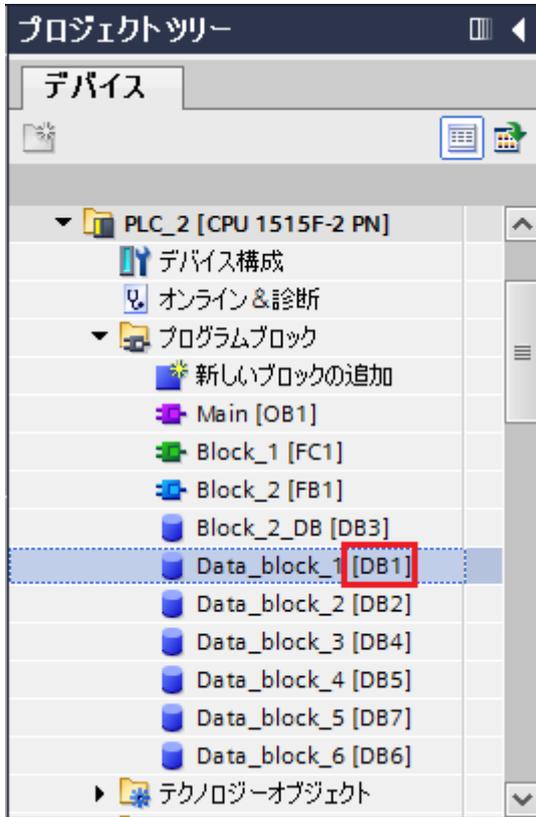
使用例

```
CCaoVariable variable = ctrl.AddVariable(  
    "VAR1", "DBNo=1, Type=DINT, Address=0, Elem=0");
```

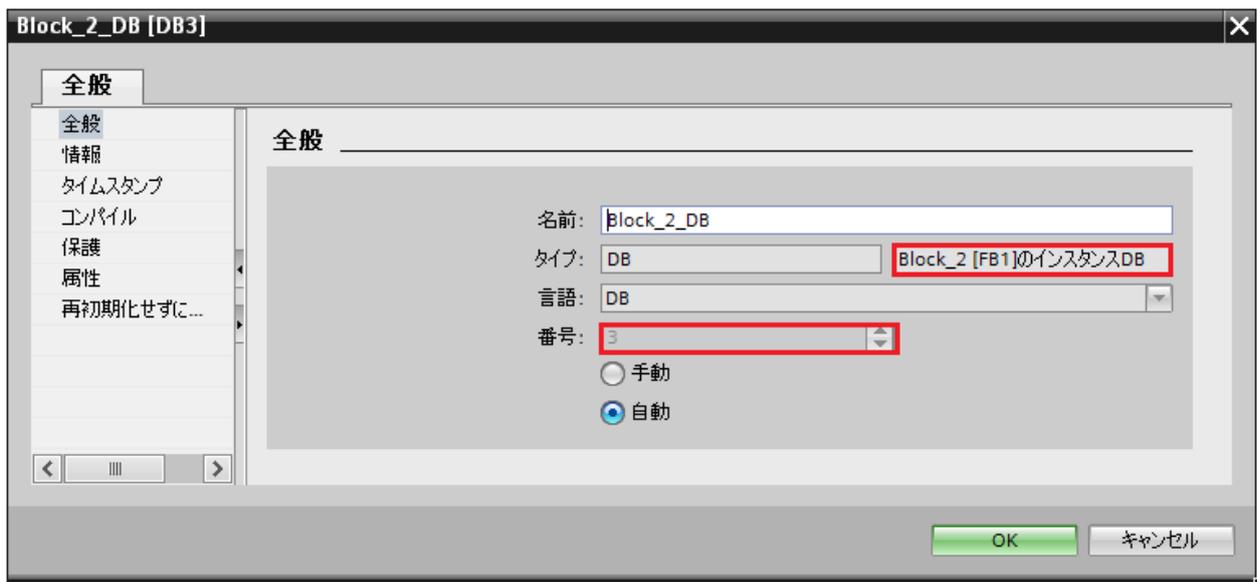
2.4.2.1. DBNo オプション

TIAPortal のデータブロック名の右側に記載されている番号を指定してください。

下図の例では 1 を指定します。



インスタンスデータブロックはプロジェクトツリー上では確認できませんので該当のデータブロックを[右クリック] - [プロパティ]を選択して確認してください。



2.4.3. GaoVariable::get_Value メソッド

AddVariable で指定した値の読み込みを行います。

使用例

```
object value = variable.Value;
```

2.4.4. GaoVariable::put_Value メソッド

AddVariable で指定した値の書き込みを行います。

Type に CharArray および WCharArray を指定した場合は文字列か 1 要素の文字列の配列を指定する必要があります。

Type に CharArray および WCharArray 以外を指定した場合は Elem で指定した要素の配列を指定する必要があります。

ただし、Elem が 0 または 1 の場合は直接値が指定できます。

使用例

```
variable.Value = "ABC";
```

3. Variable の Type の詳細

3.1. Bool

Bool はビット単位の値の読み込み/書き込みを行います。

各パラメータは以下の意味になります。

3.1.1. Address

ピリオドの左側が読み込み開始アドレス, 右側が読み込み開始アドレスのビットオフセットを表します。

3.1.2. Elem

読み込む値の数をビット単位で指定します。

3.1.3. 注意

ビットオフセットが 0 以外であるか, Elem が 8 の倍数ではない場合は影響範囲のデータをバイト単位で読み込んだ後に指定位置を更新して書き込みを行います。

このため, 書き込みを行わないビットでは読み込みから書き込みの間に行われた変更が失われる可能性があります。

3.2. Char, Byte, Int, Word, DInt, DWord, LInt, LWord, Real, LReal

Char, Byte, Int, Word, Dint, DWord, LInt, LWord, Real, LReal はビッグエンディアンのメモリイメージの読み込み/書き込みとなり, Elem は各型の大きさ単位の要素数となります。

各型の意味は以下の通りです。

Type の値	型のタイプ	型の大きさ(バイト)	符号
Char	整数	1	あり
Byte	整数	1	なし
Int	整数	2	あり
Word	整数	2	なし
DInt	整数	4	あり
DWord	整数	4	なし
LInt	整数	8	あり
LWord	整数	8	なし
Real	実数	4	-
LReal	実数	8	-

3.3. String, Wstring

String および WString は S7 シリーズ上で String および WString として定義したメモリを文字列として読み込み/書き込みを行うための Type です。

S7 シリーズで String および WString は特殊なフォーマットで記録されていますので、S7 シリーズ上で String および WString で宣言されている領域以外にアクセスする場合は String および WString は使用しないでください。

3.4. CharArray, WCharArray

CharArray および WCharArray は Char および Int の配列を文字列として取り出すための Type です。

3.4.1. 読み込み

CharArray は Elem バイト分を読み込み、ASCII 文字列として文字列化します。

WCharArray は Elem * 2 バイト分を読み込み、ビッグエンディアンの Unicode として文字列化します。

3.4.2. 書き込み

CharArray は、設定された値を ASCII 文字列としてバイト配列に変換して書き込みを行います。

WCharArray は設定された値をビッグエンディアンの Unicode としてバイト配列に変換して書き込みを行います。

4. 機種ごとの制約事項

4.1. アクセスできる DataType の制約

本プロバイダでは使用しているライブラリの制約上、機種によってアクセスできる **DataType** が制限されています。

機種ごとにアクセスできる **DataType** は以下の通りです。

アクセスできない **DataType** を指定した場合、変数の追加には成功しますが値の読み込み/書き込み時にエラーが発生します。

DataType	対応するアドレス	S7-300	S7-1500
DataBlock	DB	○	○
Input	I, E	○	○
Output	Q, A	○	○
Memory	M	○	○
Counter	C, Z	読み込み可 書き込み不可	×
Timer	T	読み込み可 書き込み不可	×

4.2. 指定できる値の型の制約

- ◆ S7-300 では型の指定で **WString** を指定すると値の読み込み/書き込み時にエラーが発生します。

5. ライセンス表記

5.1. S7.Net+

The MIT License (MIT)

Copyright (c) 2009 Jürgen Schildmann

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

付録A. アドレス指定例

コントローラ上に以下に挙げる様な定義がされている場合に、各データへアクセスするためのアドレスの指定例を以下に示します。

[TIAPortal で定義したデータブロックデータ]

Data_block_1				
	名前	データタイプ	オフセット	開始値
1	▼ Static			
2	item_DINT	DInt	0.0	1500
4	item_TEST_BOOL_0	Bool	8.0	false
11	item_TEST_BOOL_7	Bool	8.7	false
12	item_CHAR	Char	9.0	'a'
13	item_BYTE	Byte	10.0	16#88
14	item_INT	Int	12.0	16#8123
15	item_WORD	Word	14.0	16#8456
16	item_REAL	Real	16.0	-1.234
17	item_STRING	String	20.0	'12345678902...
18	item_DWORD	DWord	276.0	16#12345678
22	item_LINT	LInt	290.0	-79116035695...
23	item_LREAL	LReal	298.0	1.23456789012
24	item_LWORD	LWord	306.0	16#12345678...
25	item_UINT	UInt	314.0	65535
26	item_UDINT	UDInt	316.0	4294967295
27	item_ULINT	ULInt	320.0	1234567890123
28	item_SINT	SInt	328.0	-128
29	item_USINT	USInt	329.0	255
30	item_WCHAR	WChar	330.0	WCHAR#' '
31	item_WSTRING	WString	332.0	WSTRING#" "
34	▶ item_BYTES	Array[0..10] of Byte	858.0	
40	▶ item_LINTs	Array[0..10] of LInt	1046.0	
49	▶ item_WCHARs	Array[0..10] of WChar	4304.0	

- 例1) BOOL 型定義 (item_TEST_BOOL_0)
DBNo=1, Type=BOOL, Address=8.0
- 例2) BOOL 型定義 (item_TEST_BOOL_7)
DBNo=1, Type=BOOL, Address=8.7
- 例3) Byte 型定義 (item_BYTE)
DBNo=1, Type=BYTE, Address=10
- 例4) Int 型定義 (item_INT)
DBNo=1, Type=INT, Address=12
- 例5) DInt 型定義 (item_DINT)
DBNo=1, Type=DINT, Address=0
- 例6) DWord 型定義 (item_DWORD)
DBNo=1, Type=DWORD, Address=276
- 例7) Real 型定義 (item_REAL)
DBNo=1, Type=REAL, Address=16
- 例8) LInt 型定義 (item_LINT)
DBNo=1, Type=LINT, Address=290
- 例9) ULInt 型定義 (item_ULINT)
DBNo=1, Type=LWORD, Address=320
- 例10) LReal 型定義 (item_LREAL)
DBNo=1, Type=LREAL, Address=298
- 例11) Char 型定義 (item_CHAR)
DBNo=1, Type=CHAR, Address=9
DBNo=1, Type=CharArray, Address=9
- 例12) WChar 型定義 (item_WCHAR)
DBNo=1, Type=WORD, Address=330
DBNo=1, Type=WCharArray, Address=330
- 例13) String 型定義 (item_STRING)
DBNo=1, Type=String, Address=20
- 例14) WString 型定義 (item_WSTRING)
DBNo=1, Type=WString, Address=332
- 例15) Byte 型配列(11 要素) (item_BYTES)
DBNo=1, Type=BYTE, Address=858, Elem=11
- 例16) LInt 型配列(11 要素) (item_LINTs)
DBNo=1, Type=LINT, Address=1046, Elem=11
- 例17) WChar 型配列(11 要素) (item_WCHAR)
DBNo=1, Type=WORD, Address=4304, Elem=11
DBNo=1, Type=WCharArray, Address=4304, Elem=11

付録B. 文字列型へのアクセスについて

文字列型(String, WSTRING)は図 5-1 のような構造になっています(WSTRING 型は各要素が 2 バイトずつになっている).

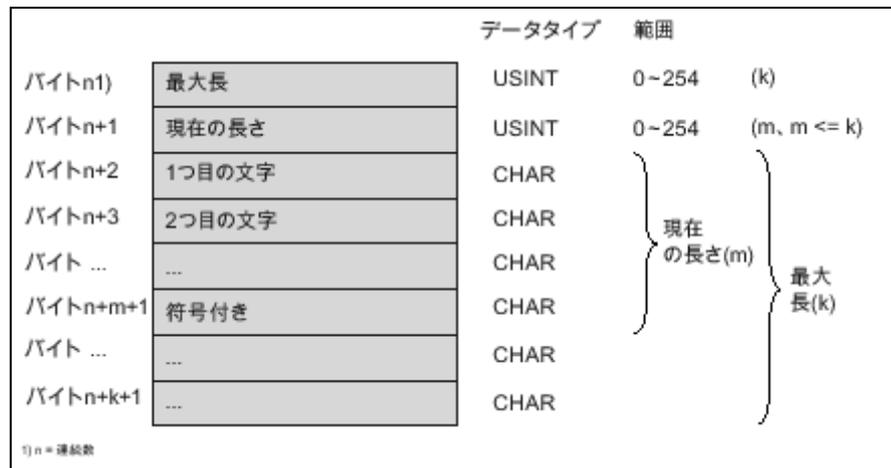


図 5-1 STRING タグの構造

put_Value による書き込み時は「最大長」を一度取得し、そのサイズを超えないように実データ部分を書き込みます。書き込み成功後、「現在の長さ」を書き込みます。

get_Value による読み出し時は「現在の長さ」を取得し、そのサイズ分実データ部分を読み出します。

付録C. リモートパートナーからのアクセス許可について

S7-1500 シリーズ¹では[PUT/GET 通信におけるリモートパートナーからのアクセス許可(...)]オプションが無効になっています。

無効の状態では本プロバイダからの通信ができませんので有効に設定します。

図 5-2 の「保護およびセキュリティ²」からアクセスレベルを「フェールセーフを含むフルアクセス(保護なし)」に設定し、「リモートパートナーからの PUT/GET 通信によるアクセスの許可」にチェックを入れます。

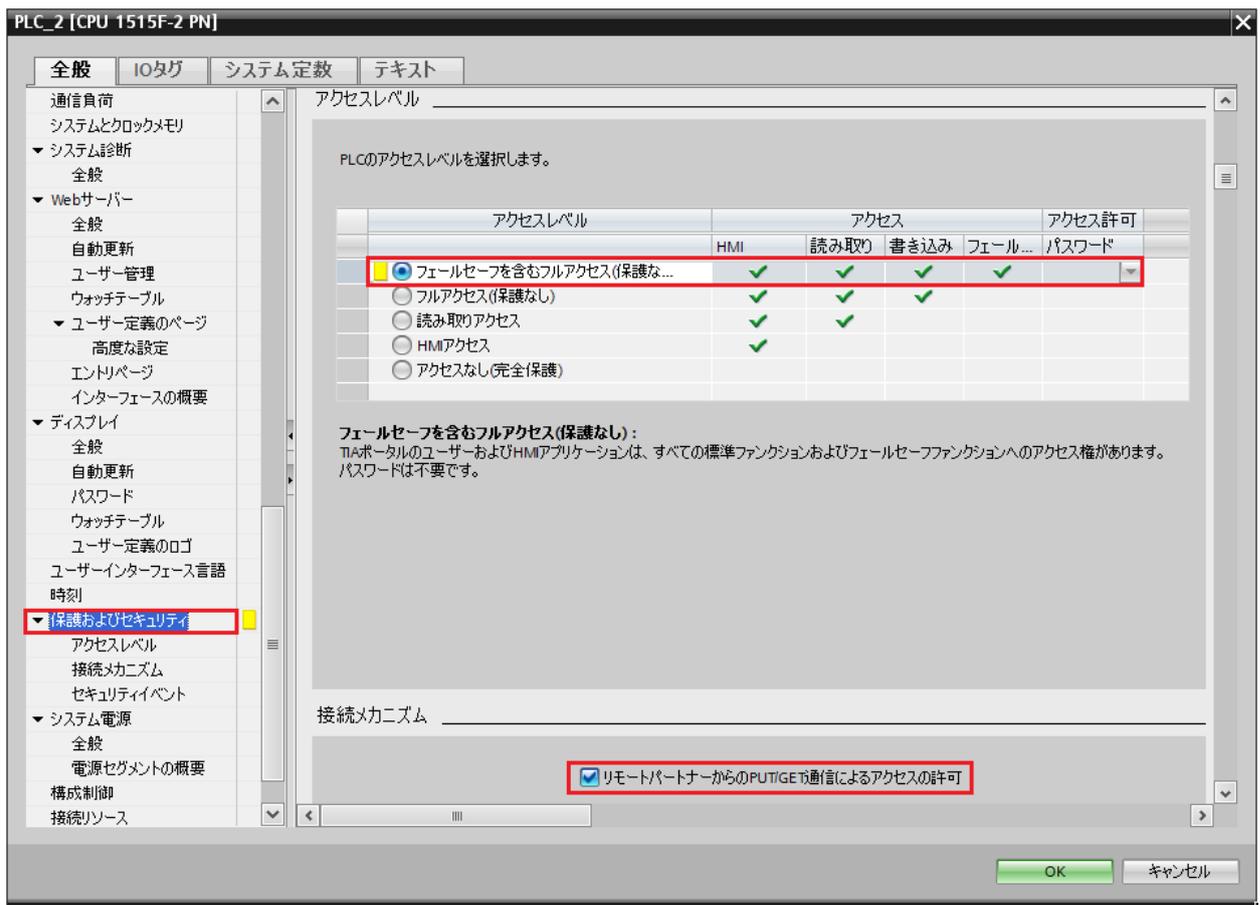


図 5-2 保護およびセキュリティ

¹ 他機種でも設定が無効になっている機種があるかもしれません。その場合は、同様の方法で有効に設定してください。

² [対象の PLC]を右クリックして[プロパティ]を選択することで設定画面を表示することができます。