



Device**Plugin**

TwinCAT plugin

ADS communication wrapper

Revision 2020.08.10

Table of Contents

1	Description	3
2	Plug-in Commands	3
2.1	*idn? (Plugin indentification)	3
2.2	open (Open device)	3
2.3	close (Close device)	3
2.4	get (Gets value from TC variable)	4
2.5	getstring (Gets string value from TC variable)	4
2.6	set (Sets value to named symbol in TwinCAT)	5
2.7	setstring (Sets value to TC variable)	5

1 Description

Plug-in provides a simplified reading and writing values over ADS library.

<i>TwinCAT</i>	Realtime software running as Windows kernel driver emulating PLC(s) which makes the control of devices over LAN interface.
<i>ADS library</i>	Provides communication channel with TwinCAT system. Communication over this library is not realtime.
<i>Symbol Name</i>	A variable defined in running TwinCAT system.
<i>AMS NetID</i>	Unique name for single TwinCAT device or service.
<i>AMS Port</i>	Any device / service in TwinCAT can has a port value for purpose of external writing or reading.

2 Plug-in Commands

2.1 *idn? (Plugin identification)

```
*idn?
```

Gets the plug-in identification string.

Parameters

No parameters.

Return value

The identification string in standard format "<company>, <product/name>, <serial-no>, <version>".

2.2 open (Open device)

```
open: <port>: { <amsNetId> }
```

Opens connection over ADS communication with given AMSNetID and application service port. TwinCAT system

Parameters

port	[number]	Service port in target TwinCAT application. Default: 851 (default plc port)
amsNetId	[string]	AMS net id. It depends on TwinCAT settings. Default: no param means the current local ams id

Return value

No return value.

2.3 close (Close device)

```
close
```

Closes the ADS communication.

Parameters

No parameters.

Return value

No return value.

2.4 get (Gets value from TC variable)

```
g: <symbolName>
get: <symbolName>
```

Gets the value from the named symbol (variable) in running TwinCAT application.
For get the **one-dimensional array** value symbol name must end with array index specifier "SymName [<index>]". Array index starts from 1 in TwinCAT apps

Parameters

symbolName	[string]	Existing symbol name in the application. For string/wstring type use getstring function.
------------	----------	---

Return value

value	[number]	Gets a numeric value. Booleans are interpreted as number 0 or 1.
-------	----------	---

Examples

```
get: GVL.bEnable
Returns 0 when bEnable is boolean in target application and have FALSE state.
```

```
get: "GVL.FieldTriggers[ 5] "
Returns value from fifth value from array FieldTriggers.
```

2.5 getstring (Gets string value from TC variable)

```
gstr: <symbolName>
getstring: <symbolName>
```

Gets the value from the named symbol (variable) in TwinCAT application.
This function does not support array of strings.

Parameters

symbolName	[string]	Existing symbol name in the application. Must be defined as string or wstring.
------------	----------	---

Return value

value	[number]	Returns string value.
-------	----------	-----------------------

Examples

```
gstr: GVL.bStringVar
Returns string value from TC variable.
```

2.6 set (Sets value to named symbol in TwinCAT)

```
s: <symbolName>: <value>
set: <symbolName>: <value>
```

Sets the value to the named symbol (variable) in running TwinCAT application.

For set the **one-dimensional array** value symbol name must end with array index specifier "SymName [<index>]". Array index starts from 1 in TwinCAT apps.

Parameters

symbolName	[string]	Existing symbol name in the application.
value	[number]	A value to write. Accepts only numbers now. For booleans use numbers 0 or 1.

Return value

No return value.

Examples

```
set: GVL.bEnable: 1
```

If bEnable is boolean in target application, set their value to TRUE.

```
set: "GVL.FieldTriggers[1]": 1,5647
```

Store number 1,5647 into first array index.

2.7 setstring (Sets value to TC variable)

```
sstr: <symbolName>: <value>
setstring: <symbolName>: <value>
```

Sets the string value to the named symbol (variable) in running TwinCAT application.

This function does not support array of strings.

Parameters

symbolName	[string]	Existing symbol name in the application. Must be defined as string or wstring.
value	[number]	A string value to write. TC string can be defined as maximum length string (variable : string[<size>]). If try to write string with more characters, function fails.

Return value

No return value.

Examples

```
sstr: GVL.sValue: "Test"
```

Write "Test" into TC variable GVL.sValue.