Volume

2

BROSIS

Engineering GmbH · Software und Industriesteuerungen

Commander

BROSIS – Commander Server

BROSIS-ENGINEERING GMBH

BROSIS Commander

Edition 3.2.1 dated 18/08/2008

Am Kirchbühl 9 • D-88099 Neukirch
Phone (+49) 07528/9516 10 • Fax (+49) 07528/9516 21

Table of Contents

Chapter

1

Introduction

About the Server of BROSIS-Commander.

The BROSIS-Commander – Server ensures communication with the control unit. It sends variables and informs the User interface(s) about changes, so that the new contents of the variables can be displayed. The use of Client-Server technology enables several user interfaces on different computers to access one server and thus, one PLC. Below is an overview of the tasks assumed by the user interface.

Task	Description
Register Variables	The user interface can register PLC variables to the server or, respectively, unregister variables in order to obtain information on changes in the variables or write values into a variable.
Protocol	Protocol entries are logged by the server. The contents of the protocol can be parameterised via the project definition.
Article Administration	The server makes the Article Data available to the user interfaces and PLC. More detailed information is given in the Project Definition Manual.

Installation

An installation routine is not provided for Brosis-Commander. All you have to do is copy the delivered files into an empty directory and start the programmes, BROSISCommander_Srv.exe, and BrosisCommander.exe.

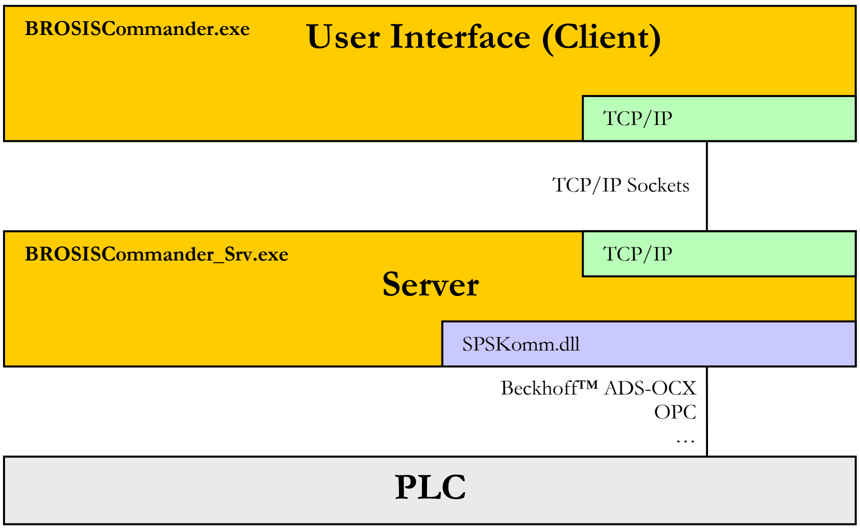
Due to the structure of the programme we recommend using the directory structure described below:

Directory	Content
Main Directory	Main directory created by you.
\Client	The directory includes all the files and programmes required by the Client. These are:

\Client\Bilder	Includes the images to be accessed by the user interface, which are not firmly embedded in the project.
\Server	The directory includes all the files and programmes required by the Server.
\Server\TypeInfo	Includes the Article Data and Project File of the server (Server.cfg)
\Server\Protocols	Includes the protocol files.

Server Structure

Below is an illustration of the internal Server structure, as well as of its connections to the user interfaces (Clients) and PLC. In addition, a brief description of the individual components is given.



- **BROSISCommander.exe**, is the actual User Interface.
- **ServerKomm.dll**, establishes the connection to the server. Using the standard ServerKomm.dll, this is effected via TCP/IP sockets.
- **BrosisCommander_Srv.exe**, is the server.
- **SPSKomm.dll**, establishes the connection to the PLC. Several versions are available, e.g. for Beckhoff™: ADS-OCX, OPC, …

Chapter
2

Files and Parameterisation

Description of the required files and Server Parameterisation.

Below please find a list and explanation of all the files required by the server:

File	Content
\Server	
BROSISCommander_Srv.exe	Executable programme file of the Server.
BROSISCommander_Srv.ini	INI file for Server Parameterisation. The possible entries are described below in this Chapter.
SPSKomm.dll	Module for communication with the PLC. By exchanging this module it is possible to access many different control units. However, please note that the syntax of the variable names depends on the access path. **Note:** If desired, this DLL also allows access to various different control units.
SPSKomm*.dll	Various communication files for Multi-PLC –SPSKomm.dll.
BorlndMM.dll	Library for memory administration.
Benutzer.xml	Includes the login data for users (name and password in XML format). This file is generated automatically!
BenutzerTmp.xml	Saves the User file last saved. This file is generated automatically!
Fenster.ini	Includes the settings for server windows (position, size….). This file is generated automatically; after deleting the file, the standard settings are used again.
Registrierung.txt	Includes the server registration data. The registration is linked to the MAC address of the network card. If no network card is available, the ID of the hard disk for drive

	C: will be used. **Note:** You may insert comment rows within a registration file by entering # as the first sign of a row.
Logo.jpg	If you do not wish the BROSIS logo to appear after the start, you may also embed your own logo. Please note that the pixel size is 430 x 101. **Note:** If the User Interface is not registered, the BROSIS logo will appear again.
BROSISCommander_SrvTexte.xml	Language texts for language selection.
Einstellungen.xml	Settings for the Multi-PLC –SPSKomm.dll.
\Server\TypeInfo	
Server.cfg	Server configuration file. This file can be generated by means of the project definition. After generating this file the server has to be restarted.
Typen.xml and/or Typen.dat	Includes Article Data (either in XML or binary format). By saving this file you will save your complete type data. To install a previously saved type file, please proceed as follows: • Terminate the server. • Install the type file. • Start the server.

Ini-File Standard Input

Server parameterisation is effected via an INI file (BrosisCommander_Srv.ini). Besides permanently existing entries there are special entries, which are dependent on the DLLs used (e.g. access to the PLC (OPC, ADS-OCX, …).

BROSISCommander_Srv.ini

Input	Description
[Server]	
Port	Number of the port the TCP/IP socket server has to respond to. This input has to be identical with that of the Client. **Note:** Besides the port indicated here, Port+1 is also used

	for communication :
BlockSendeIntervall	If an interval is entered here (in milliseconds), the server will optimise the transfer of variables to the clients. In this case, “BlockSendInterval” will collect – in milliseconds – the variables that have changed, and then transfer them as a one data block.
[Einstellungen]	
ServerName	Normally there is a programme-internal “lock” to prevent the server from being accessed for several times. However, if this should be necessary, you may assign different names to the servers. In addition to this change, you have to define another port for the server.
Silent	Possible values are 0 or 1 (Default). The entry determines if a message is to be returned at unauthorised multiple start of the server (0) or not (1).
ServerConfigName	Indicates the name and path of the configuration file. The path entry can also be relative to the server directory. If no entry exists, the system will use “Server.cfg”. **Example** [Settings] ServerConfigName = .\TypInfos\Server.cfg
TypVerzeichnis	Indicates the path of the Article Data Files. The path entry can also be relative to the server directory. If no entry exists, the programme will use “„.\TypeInfo“. **Example** [Settings] TypeDirectory = .\TypeInfo
TypXML	Possible values are 0 (Default) or 1. The entry defines if article data has to be stored in the XML format (1) or binary format (0). The advantage of the XML format is that data can be stored in a legible format – however, this will affect the file size.
Registrierung	Name and path of the registration file. If no entry exists, the programme will use “Registrierung.txt“.
UnicodeProtokoll	Possible values are 0 (=Unicode) or 1 (=ASCII). The entry indicates if the protocols have to be stored in the UNICODE format or in the normal ASCII format.
Debug	Possible values are True or False. The entry determines if the Debug menu has to be displayed. This entry can

	also be changed during server operation. In this case, you may display or hide the menu item without restarting the server.
Extras	Possible values are True or False. The entry defines if the Extras menu has to be displayed. This entry can also be changed during server operation. In this case, you may display or hide the menu item without restarting the server.
[Sprache]	
ServerLaeuftText	Defines the text shown if there is an attempt to start the server although it is already running. If this entry does not exist, the programme will use the standard message “Server already running”.
SprachCode	Number of the language used for starting up the server.
[Information]	
Zeile_1	Allows you to enter the information shown at the start. If nothing is entered here, the following information will be given: BROSIS Engineering GmbH Am Kirchbühl 9 D-88099 Neukirch Phone: (+49) 07528/9516-10 Email: info@brosis-engineering.de
Zeile_2	
Zeile_...	

Input for PLC Communication

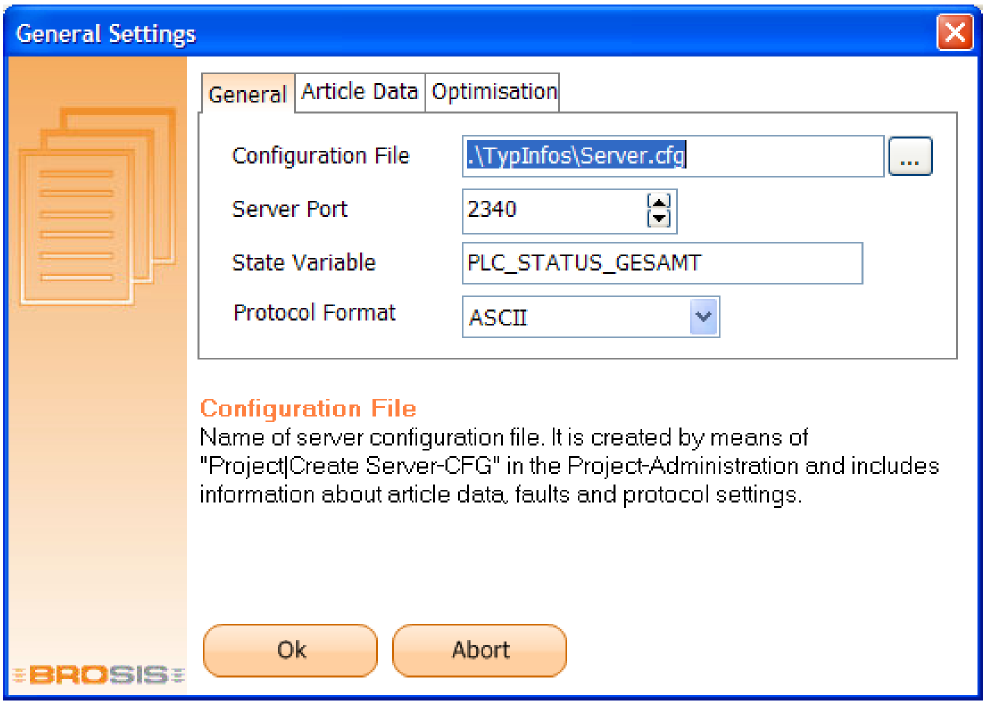
Which type of communication is supported by the SPSKomm.dll currently being used, can be determined via Data Properties (Dateieigenschaften) (Version Info) or simply by dragging the mouse over the file in the Explorer. Please wait for a few moments – the version information will be displayed in a dialogue.

Multi-Communication DDL

This communication DLL allows access to several (even different) DLLs. The accesses are distinguished by the PLC number. The setting is menu-guided via the Extras menu.

Extras Menu – General

This menu is used to perform general server settings. The entries are described in the Info box.



Configuration File .\TypInfos\Server.cfg

Server Port 2340

State Variable PLC_STATUS_GESAMT

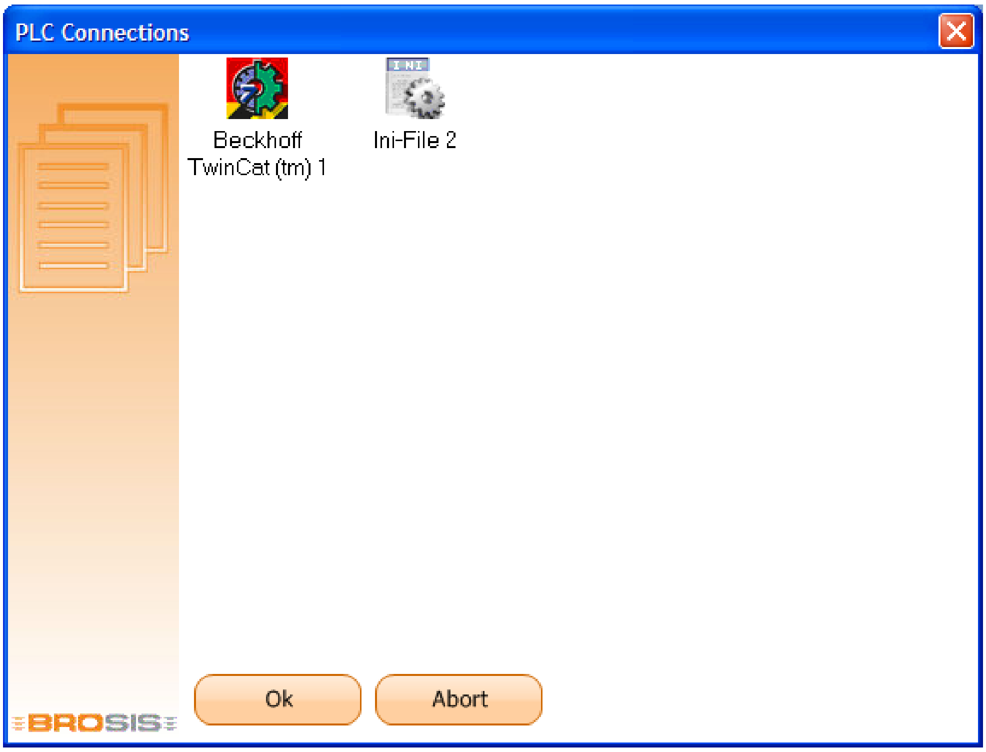
Protocol Format ASCII

Configuration File

Name of server configuration file. It is created by means of "Project|Create Server-CFG" in the Project-Administration and includes information about article data, faults and protocol settings.

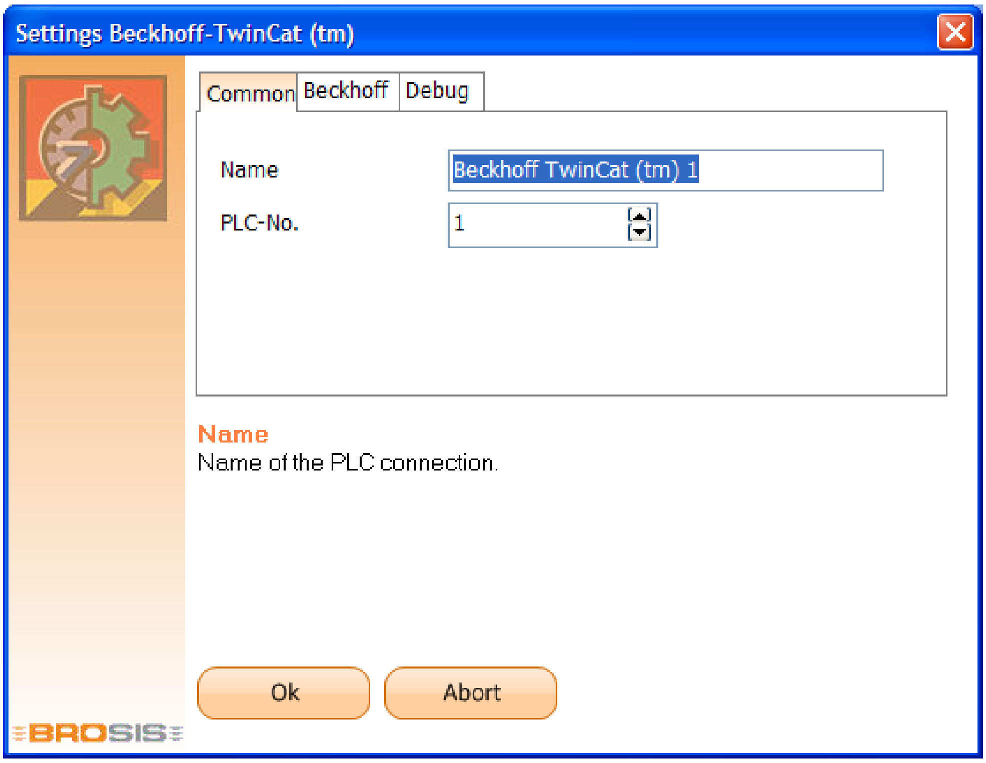
Extras Menu | PLC Connections

Allows configuring the connections to the control units. Using the right mouse button, connections can be created, changed, or deleted.



Using the right mouse button, connections can be added, edited and deleted. Which communications are offered is determined by the SPSKomm*.dll – files, which are located in the directory.

Example of a Beckhoff connection:



Name: Beckhoff TwinCat (tm) 1

PLC-No.: 1

Name

Name of the PLC connection.

The individual entries are described in the Info box !

The communication DDLs described below are still supported for compatibility purposes. All these connections can be established by the multi-communication DDL.

Beckhoff™ ADS Communication

Enables access to a Beckhoff-PLC (not BKxxxx) (max. 2 100 variables can be entered).

Input	Description
[Beckhoff]	
ServerNetId	AMS-Net-ID of TwinCat Soft-PLC to which the connection is to be established. This entry may be omitted if you wish to connect to the local PLC.
ServerPort	AMS-Server-Port of TwinCat Soft-PLC to which the connection is to be established. The value is 801 for PC controls (for the first runtime).
Zykluszeit	Cycle time, in milliseconds, for importing the entered variables (Beckhoff™ ADS internal), provided that the standard value is entered in the Project Definition. This value should be at least 200 ms (depending on the number of variables entered in this way).

The variables are accessed as shown in the Watch Window of the TwinCat-PLC-Controls.

Variable Type	Syntax
Global Variable	.<Variablename>
Global Structure Variable	.<Strukturvariable>.<Variablenname>
Global Array Variable	.<Variablenname>[<Index>]
Programme Variable	<Programmname>.<Variable>

Beckhoff™ Multi ADS-Communication

Allows access to several (max. 4) Beckhoff-PLC (not BKxxxx). A maximum of 700 variables can be entered for each PLC.

Input	Description
[Beckhoff]	

Zykluszeit	Cycle time, in milliseconds, for importing the registered variables (Beckhoff™ ADS internal), provided that the standard value is entered in the Project Definition. This value should be at least 200 ms (depending on the number of variables entered in this way).
[Server_1]	
ServerNetId	AMS-Net-ID of TwinCat Soft-PLC to which the connection is to be established. This entry may be omitted if you wish to connect to the local PLC.
ServerPort	AMS-Server-Port of TwinCat Soft-PLC to which the connection is to be established. The value is 801 for PC controls (for the first runtime).
Nummer	PLC number by means of which this PLC is accessed during the parameterisation.
[Server_2]	
…	See Section [Server_1]
[Server_3]	
…	See Section [Server_1]
[Server_4]	
…	See Section [Server_1]

The syntax of the variables is the same as for the access to a PLC. However, the PLC number has to be entered in addition.

OPC Communication

Enables access to any desired PLC. Access is via OPC, i.e. you will need an additional OPC server enabling access to the desired PLC.

Input	**Description**
[OPC]	
OPC-Server	Name of the OPC server to be accessed. You will find this name in the documentation of the OPC server.
UpdateRate	Update rate, in milliseconds, for importing the variables.
NurString	If you wish the OPC server to provide data in a text format only, you have to enter 1.
ZyklischLesen	Enter 1 for cyclic reading, if you do not want event-controlled OPC server operation.
LeseZyklus	Time interval for cyclic reading, in milliseconds.

The syntax of the variables is determined by that of the OPC server.

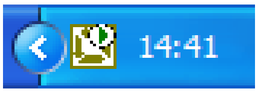
Chapter

3

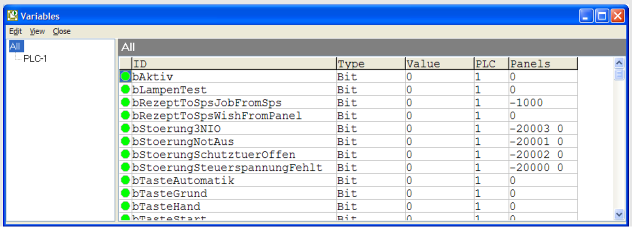
Programme Operation

Access of Operation of the BROSISCommander-Server.

The BROSISCommander is executed after starting the icon in the task bar. Clicking this icon will open a menu by means of which the server can be operated. Which menu items are available can be determined via the access parameters. The graph below shows the task bar icon of the BROSIS-Commander server. Drag the mouse over the icon and wait for a few seconds until the name of the server is displayed. This is a useful function when running several servers at the same time.

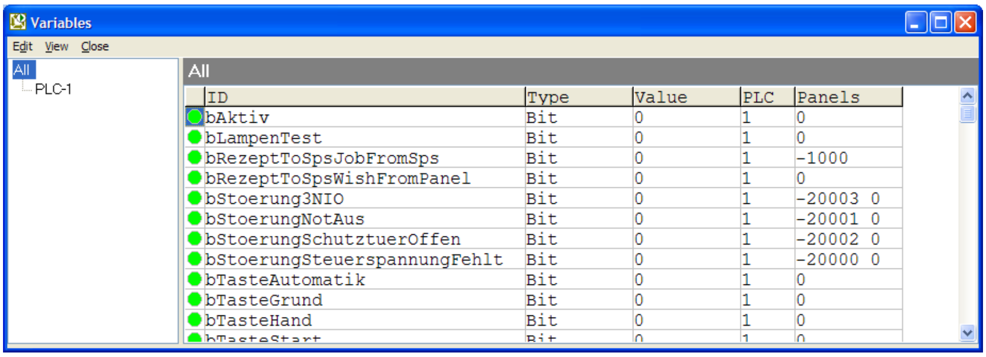


The Menu

<table>
<tr><th>Menu Item</th><th>Description</th></tr>
<tr><td>Extras</td><td>This menu includes special dialogues for the settings of the communication file.</td></tr>
<tr><td colspan="2"><b>Debug</b></td></tr>
<tr><td>Log Output</td><td>Shows the Log Output window. This window can be used, for example, to obtain more detailed information about problems that may have occurred when entering variables.</td></tr>
<tr><td>Registered Variables</td><td>Opens an overview of all registered variables. The following information concerning the entered variables will be displayed:
• Connection Status:
Red = Not connected
Yellow = Connection is built up
Blue = Registered for Write Only
Green = Connected


<table>
<tr><th></th><th>ID</th><th>Type</th><th>Value</th><th>PLC</th><th>Panels</th></tr>
<tr><td>●</td><td>bAktiv</td><td>Bit</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>●</td><td>bLampenTest</td><td>Bit</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>●</td><td>bRezeptToSpsJobFromSps</td><td>Bit</td><td>0</td><td>1</td><td>-1000</td></tr>
<tr><td>●</td><td>bRezeptToSpsWishFromPanel</td><td>Bit</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>●</td><td>bStoerung3NIO</td><td>Bit</td><td>0</td><td>1</td><td>-20003 0</td></tr>
<tr><td>●</td><td>bStoerungNotAus</td><td>Bit</td><td>0</td><td>1</td><td>-20001 0</td></tr>
<tr><td>●</td><td>bStoerungSchutztuerOffen</td><td>Bit</td><td>0</td><td>1</td><td>-20002 0</td></tr>
<tr><td>●</td><td>bStoerungSteuerspannungFehlt</td><td>Bit</td><td>0</td><td>1</td><td>-20000 0</td></tr>
<tr><td>●</td><td>bTasteAutomatik</td><td>Bit</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>●</td><td>bTasteGrund</td><td>Bit</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>●</td><td>bTasteHand</td><td>Bit</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>●</td><td>bTasteStart</td><td>Bit</td><td>0</td><td>1</td><td>0</td></tr>
</table></td></tr>
<tr><td>Save Type Infos</td><td>Allows saving the type information (article data) in a format that is currently not used for saving. For example, if data is saved in the binary format, this menu item will transfer the data to XML format.</td></tr>
<tr><td>Connection</td><td>Provides a “small statistics” on the communication between the server and user interface.</td></tr>
<tr><td>Information</td><td>Opens a dialogue showing the version number and registration data.</td></tr>
<tr><td>Languages</td><td>Enables changing the programme language.</td></tr>
<tr><td>Terminate</td><td>Terminates the server after a prompt.</td></tr>
</table>

Example: Dialogue with entered/sent variables:



	ID	Type	Value	PLC	Panels
●	bAktiv	Bit	0	1	0
●	bLampenTest	Bit	0	1	0
●	bRezeptToSpsJobFromSps	Bit	0	1	-1000
●	bRezeptToSpsWishFromPanel	Bit	0	1	0
●	bStoerung3NIO	Bit	0	1	-20003 0
●	bStoerungNotAus	Bit	0	1	-20001 0
●	bStoerungSchutztuerOffen	Bit	0	1	-20002 0
●	bStoerungSteuerspannungFehlt	Bit	0	1	-20000 0
●	bTasteAutomatik	Bit	0	1	0
●	bTasteGrund	Bit	0	1	0
●	bTasteHand	Bit	0	1	0
●	bTasteStart	Bit	0	1	0