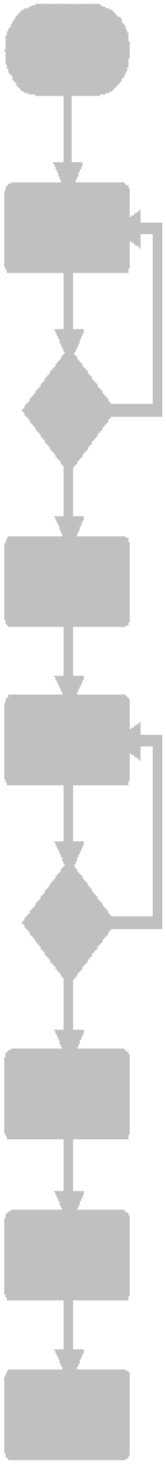

JetControl 24x

Version Update

from V. 3.15 to V. 3.16

Jetter
AKTIENGESELLSCHAFT

Edition 1.00 September 2005 / Printed in Germany

Jetter AG reserves the right to make alterations to its products in the interest of technical progress. These alterations need not to be documented in every single case.

This user information and the information contained herein have been compiled with due diligence. Jetter AG shall not be liable for errors contained herein or for incidental or consequential damage in connection with the furnishing, performance, or use of this material.

Table of Contents

1 Introduction

Version Updates - Survey			
Version	**Function**	**upgraded**	**corrected**
V. 3.16	Interpreter	✓	✓
V. 3.15	Communication	✓	✓
	Registers	✓	
	Interpreter		✓
	System bus	✓	✓
	E-mail	✓	
3.14	System bus	✓	✓
	Communication	✓	
	Operating system update	✓	
	Interpreter	✓	✓
	Registers	✓	✓
	Debugger		✓
	HTTP server	✓	

2 Expansions

2.1 Display Instructions

Display_Text and Display_Register instructions can also be rerouted to character string variables or text registers. For this, device number **# 7** needs to be specified in the instruction. If device number # 0 is applied (indirect device number), 7 must be written into register 2824.

The number of the first register among the text registers, respectively the address of the character string variable, must first be set in register **2841**.

Control characters for clearing the display (register 2839; '_') and for clearing a line up to the end of the line (register 2840; '$') are removed from the character string. The registers for formatting the outpu (field length, number of decimal positions, etc.) are taken into consideration.

Register 2841: Address of the character string variable for rerouting to device # 7	
Function	**Description**
Read	Addresses of variables
Write	Set a new address
Value range	0 .. 1999 and 20000 .. 49999
Value after reset	0

2.2 String Functions

By means of the special / system functions described below, strings can be processed in the character string variable format. A description of the format can be found in the JetSym online help.
The maximum length of a string must not exceed 255 characters!
Please make sure the strings do not overlap; otherwise the results of the functions will not be unambiguous.

2.2.1 Comparing strings

With the help of special function **140**, the contents of two string variables are compared.

```
SYSTEM_FUNCTION(140, <Source Reg. No.>, <Destination Reg. No.>)
```

<Source Reg. No.>	Specifies the number of the first register of the parameter block.
<Destination Reg. No.>	Specifies the result register number for this function.

Parameter block

The function parameters are specified starting from register **<Source Reg. No.>**.

Register offset	Meaning	
0	Address of string 1	It specifies the number of the first register of the first string variable
1	Address of string 2	It specifies the number of the first register of the second string variable

Function result

The result of the function can be read out of register **<Destination Reg. No.>**.

Register Contents	Meaning
= 0	The strings are identical
< 0	String 1 is shorter than string 2
> 0	String 1 is longer than string 2

2.2.2 Find the string in another string

Special function **141** helps to check, whether string2 is in string1.

SYSTEM_FUNCTION(141, <Source Reg. No.>, <Destination Reg. No.>)

<Source Reg. No.>	Specifies the number of the first register of the parameter block.
<Destination Reg. No.>	Specifies the result register number for this function.

Parameter block

The function parameters are specified starting from register **<Source Reg. No.>**.

Register offset	Meaning	
0	Address of string 1	It specifies the number of the first register of the first string variable
1	Address of string 2	It specifies the number of the first register belonging to the second string variable, which is searched for within string 1.

Function result

The result of the function can be read out of register **<Destination Reg. No.>**.

Register Contents	Meaning
0	String 2 has not been found within string 1
1	String 2 has been found within string 1

2.2.3 Joining the strings

By means of special function **142**, the content of string variable 2 is joined with the content of string variable 1. String variable 2 is not changed.

```
SYSTEM_FUNCTION(142, <Source Reg. No.>, <Destination Reg. No.>)
```

<Source Reg. No.>	Specifies the number of the first register of the parameter block.
<Destination Reg. No.>	Specifies the result register number for this function.

Parameter block

The function parameters are specified starting from register **<Source Reg. No.>**.

Register Offset	Meaning	
0	Address of string 1	It specifies the number of the first register of the first string variable
1	Address of string 2	It specifies the number of the first register of the second string variable
2	max. register	The maximum number of registers being part of string 1 is specified

Function result

The result of the function can be read out of register **<Destination Reg. No.>**. The result register contains the number of registers occupied by the new string 1.

2.2.4 Converting register values into strings

With the help of special function **143**, the value of a register can be converted into a string and stored to a string variable. The register value will not be changed in this process.

```
SYSTEM_FUNCTION(143, <Source Reg. No.>, <Destination Reg. No.>)
```

<Source Reg. No.>	Specifies the number of the first register of the parameter block.
<Destination Reg. No.>	Specifies the result register number for this function.

Parameter block

The function parameters are specified starting from register **<Source Reg. No.>**.

Register Offset	Meaning	
0	Register number	Specifies the number of the register, the value of which is to be converted
1	Address string	Specifies the number of the first register of the string variable, to which the string is to be stored

Function result

The result of the function can be read out of register **<Destination Reg. No.>**. The result register contains the number of registers occupied by the string variable.

2.2.5 Copying strings

By means of special function **144**, the content of string variable 1 is copied into string variable 2. String variable 1 is not changed.

```
SYSTEM_FUNCTION(144, <Source Reg. No.>, <Destination Reg. No.>)
```

<Source Reg. No.>	Specifies the number of the first register of the parameter block.
<Destination Reg. No.>	Specifies the result register number for this function.

Parameter block

The function parameters are specified starting from register **<Source Reg. No.>**.

Register Offset	Meaning	
0	Register number	Specifies the number of the first register of the string variable, the value of which is to be copied
1	Address string	Specifies the number of the first register of the second string variable, to which the contents of string 1 are to be copied

Function result

The result of the function can be read out of register **<Destination Reg. No.>**. The result register contains the number of registers occupied by the string variable.

3 Eliminated Software Bugs

3.1 Display_Text

An error had crept into V. 3.15, which would stop the processing of the application program and cause bit 5 (illegal opcode in the application program) to be set in error register 2008.
The error occurred at the Display_Text instruction, when character string variables or text registers were to be displayed.
At direct specificaiton of a text, or at a display_register instruction, no errors would occur.