Jetter System Bus

Smart IOs / JX-SIO

Function Description

Article #
Edition 1.30

June 2002 / Printed in Germany

Table of Contents

1 Description 5

1.1 System Requirements 5

1.2 System Bus - Baud Rate 5

1.3 System Bus - Technical Data 6

2 Installation 7

2.1 JX-SIO Module - Settings 7

2.2 Connection to the System Bus 8

2.3 Power Supply of the JX-SIO Module 10

2.4 System Bus Cable 11

3 Software and Programming 12

3.1 Numbering Inputs and Outputs 12

3.2 Addressing the Digital Inputs and Outputs 12

3.3 Access to IOs through Overlaying of Registers 14

3.4 Registers for Analog Inputs 18

3.4.1 Reading Analog Input Values 18
3.4.2 Value Range of analog inputs 19
3.4.3 Configuring Analog Inputs 20

3.5 Analog Input Terminals - Overview 22

3.5.1 Analog Input Terminal Module IB IL AI 2/SF 22
3.5.2 Analog Input Terminal Module IB IL AI 8/SF 26
3.5.3 Temperature Terminal Module IB IL TEMP 2 RTD 36
3.5.4 Temperature Terminal Module IB IL TEMP 2 UTH 40

3.6 Registers for Analog Outputs 44

3.6.1 Writing of Analog Output Values 44
3.6.2 Value Range of analog outputs 45
3.6.3 Configuring Analog Outputs 46

3.7 Analog Output Terminals - Overview 48

3.7.1 Analog Output Terminal Module IB IL AO 2/U/BP 48

3.8 Response of Digital and Analog I/Os to Errors 49

3.8.1 Monitoring Interval 49
3.8.2 Configuring the Response of Digital Outputs to Errors 50
3.8.3 Configuring the Response of Analog Outputs to Errors 52

4 Register Description 54

4.1 System Bus Management 54
4.2 Terminal Array of the JX-SIO Module 56
5 JX-SIO Module Diagnostics 59
5.1 Diagnostic Indicators (LEDs) 59
5.2 Register Addressing 61
5.3 Diagnostic Registers with NANO CPUs 61
5.4 Diagnostic Registers with JetControl 24x 63
Calculating the Extension Size 65
Register Overview 70
Glossary 77
Tips and Tricks 78

1 Description

This document describes how to connect Smart IOs JX-SIO to the Jetter System Bus and how to operate them.
In addition to this description, the following documentation applies: Data sheets for individual terminals and the installation guideline for the INTERBUS INLINE system.

1.1 System Requirements

Smart IOs JX-SIO can directly be connected to the Jetter System Bus. All JETTER brand expansion modules can simultaneously be operated on the system bus.
The following table shows the minimum required software versions supporting operation of JX-SIOs on the Jetter System Bus.

Software Versions of Controllers	
Controller	**Minimum Software Version**
Nano-B	V2.00
Nano-C	V3.00
Nano-D	V2.00
JetControl 241	V2.00
JetControl 243	V2.00
JetControl 246	V2.00

1.2 System Bus - Baud Rate

The system bus of JETTER AG can be operated at baud rates between 125 kBaud and 1 MBaud. In general, it can be said that the higher the baud rate the shorter the maximum cable length. At the same time, the transfer rate of the system bus increases with higher baud rates, too. Therefore, for every individual application you have to decide whether the system bus should be operated with the maximum transfer rate or the maximum cable length.
The permissible baud rates of the system bus also depend on the modules connected to it.

Permissible Baud Rates					
Non-intelligent JX2 modules **Intelligent JX2 modules**	**Smart IOs JX-SIO**	**125 kBaud**	**250 kBaud**	**500 kBaud**	**1000 kBaud**
✓		✓	✓	✓	✓
	✓	✓	✓	✓	✓
✓	✓	✓			✓

1.3 System Bus - Technical Data

The maximum system bus configuration depends on several marginal data. On the one hand, each controller can accommodate only a limited number of inputs and outputs. On the other hand, the expandability of a JX-SIO is limited as well as the number of JX-SIO modules to be connected.

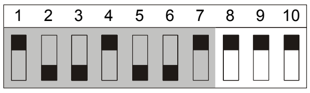
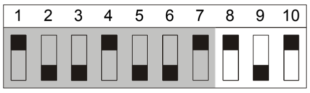
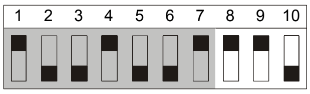
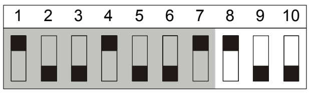
System Bus - Expandability	
Function	**Description**
Maximum amount of IOs **Once the maximum amount of IOs has been reached, all information given in this table is limited to this amount.**	The maximum amount of modules to be connected to the system bus is limited by the amount of digital inputs and outputs. In order to calculate the amount of IOs please refer to the tables given in chapter 0 Calculating the Extension Size
Maximum amount of non-intelligent JX2 modules	Nano-B : 15 Nano-C : 15 Nano-D : 23 JetControl 241 : 7 JetControl 243 : 15 JetControl 246 : 23
Maximum amount of intelligent JX2 modules	Nano-B : 3 Nano-C : 3 Nano-D : 4 JetControl 241 : 1 JetControl 243 : 3 JetControl 246 : 6
Maximum amount of JX-SIO	10
Maximum amount of digital inputs per JX-SIO module	64
Maximum amount of digital outputs per JX-SIO module	64
Maximum amount of analog inputs per JX-SIO module	12
Maximum amount of analog outputs per JX-SIO module	12
Maximum amount of terminals per JX-SIO module	63

2 Installation

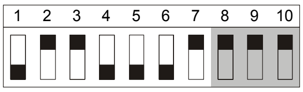
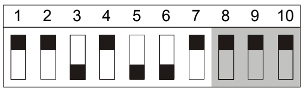
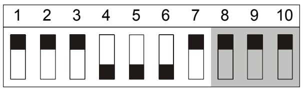
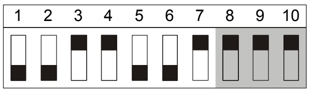
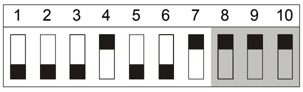
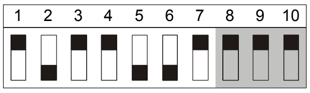
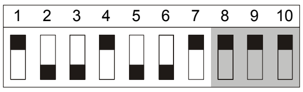
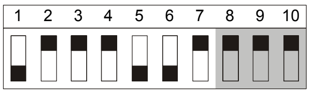
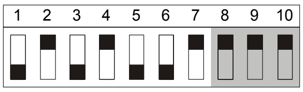
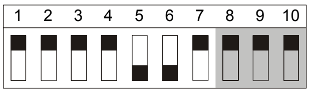
2.1 JX-SIO Module - Settings

The baud rate, as well as the module number can be set by means of switch S41. The same module number must not be assigned to two different JX-SIO modules. Moreover, all expansion modules connected to the system bus must be operated with the same baud rate.

Switch S41 – Setting the Baud Rate

Baud	S41.8 ... S41.10	Baud	S41.8 ... S41.10
1000k		250k	
500k		125k	

Switch S41 – Setting the Module Number

Module #	S41.1 ... S41.7	Module #	S41.1 ... S41.7
70		75	
71		76	
72		77	
73		78	
74		79	

2.2 Connection to the System Bus

Connecting the JX-SIO module to system bus

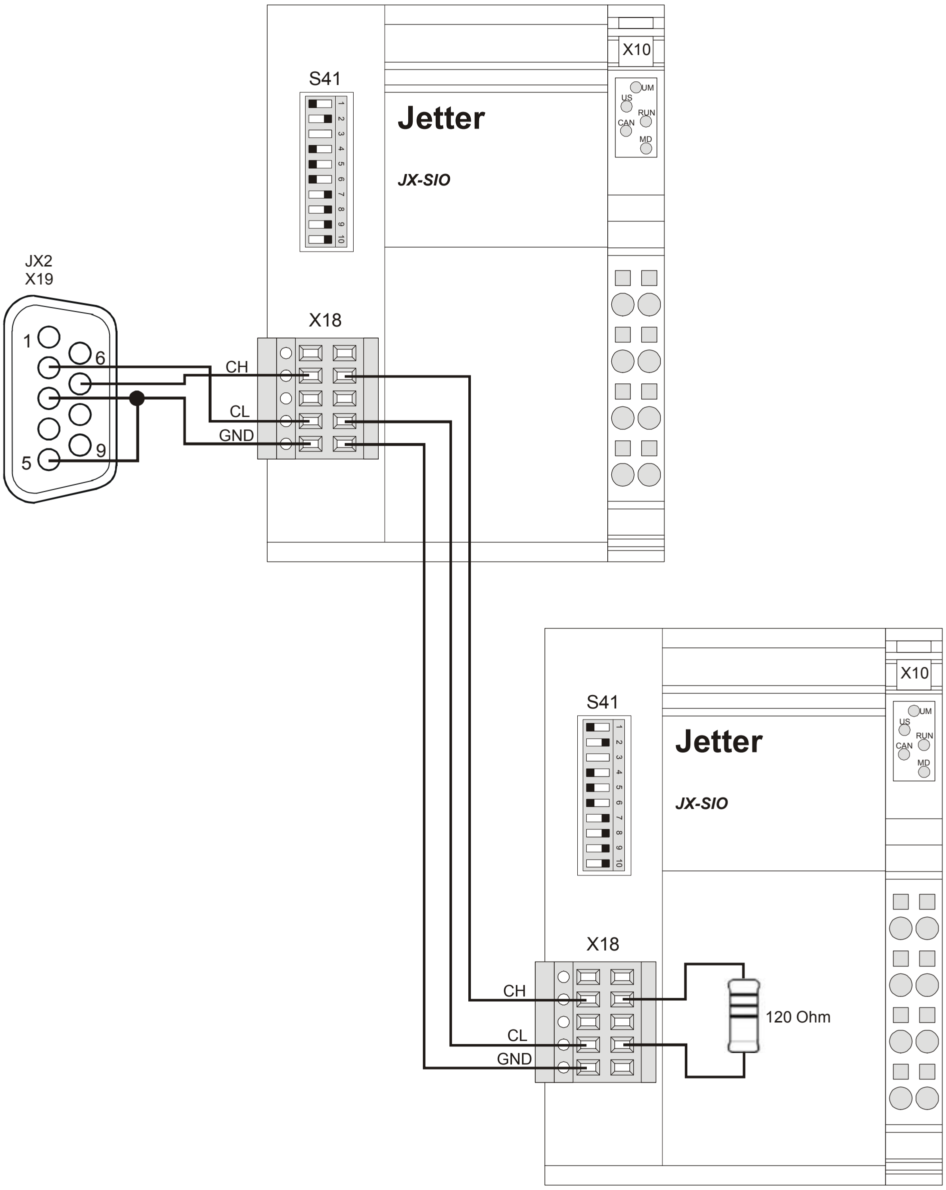


Figure: Connecting the JX-SIO to the System Bus

The JX-SIO can be connected to the end of the system bus. When connecting a module give heed to the fact that pin 3 is connected to pin 5 of connector X19 of the last intelligent or non-intelligent module.
In order to connect two JX-SIO modules a shielded cable must be used.
On the JX-SIO which is connected to system bus last, a 120 Ω resistor must be connected to the terminals CH and CL.

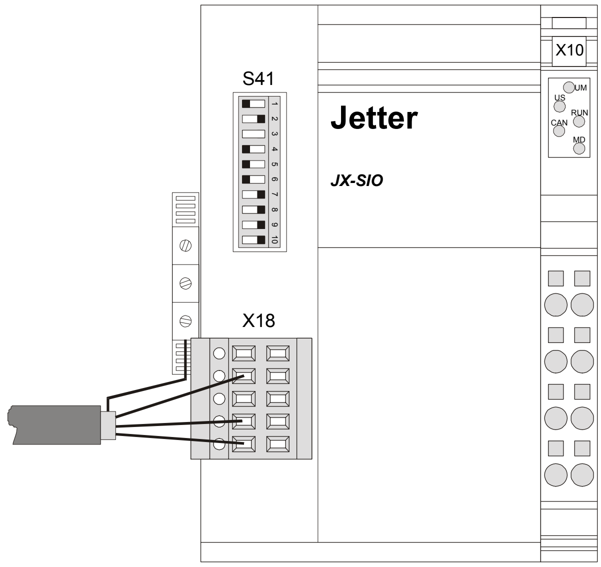


Figure: Connecting the Cable Shield

To ensure optimum disturbance suppression of the Jetter System bus cable, we recommend to directly connect the cable shield to a grounding terminal of the JX-SIO module.

2.3 Power Supply of the JX-SIO Module

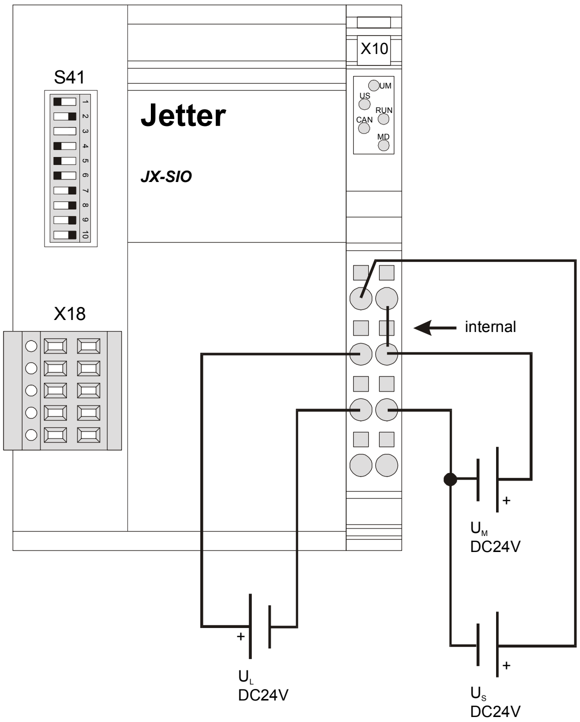


Figure: Connecting the supply voltages to the JX-SIO module

Three different supply voltages have to be connected to the JX-SIO module. The voltages U_M and U_S are for supplying the terminals. The voltage U_L is for supplying the JX-SIO module itself.

Voltages:

U_L	Internal supply voltage of the JX-SIO Module
U_M	Main voltage for supplying the terminals
U_S	Segment voltage for supplying the terminals

2.4 System Bus Cable

For the manufacture of the system bus cable the following minimum requirements apply:

System Bus Cable - Technical Data	
Core cross-sectional area	0.25 mm^2
Cable capacitance	maximum 60 pF / m
Resistivity	maximum 70 Ω / km

The maximum cable length depends on the baud rate used and number of modules connected to the bus. To the system bus cable the following rule of thumb applies: Each module connected to the bus reduces the cable length by approx. 1 m.

Permissible Cable Lengths

Baud Rate	Max. Cable Length	Max. Tap Line Length	Max. Overall Tap Line Length
1000 kBaud	30 m	0.3 m	3 m
500 kBaud	100 m	1 m	39 m
250 kBaud	250 m	3 m	78 m
125 kBaud	500 m	-	-

3 Software and Programming

3.1 Numbering Inputs and Outputs

Inputs and outputs of a JX-SIO module are numbered according to a determined scheme. Digital inputs and outputs, as well as analog inputs and outputs are numbered separately of each other.

Numbering the digital inputs

Digital inputs are numbered consecutively beginning from the JX-SIO module proceeding to the right. To the first digital input number `IN 7y01` assigned. Numbering has to be carried out without gaps.
Each terminal occupies as much inputs as specified on the data sheet under "Input Address Space".

Numbering the digital outputs

Digital outputs are numbered consecutively beginning from the JX-SIO module proceeding to the right. To the first digital output number `OUT 7y01` assigned. Numbering has to be carried out without gaps.
Each terminal occupies as much outputs as specified on the data sheet under "Output Address Space".

Numbering the analog inputs

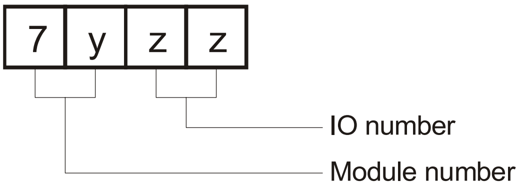
Analog inputs are numbered consecutively beginning from the JX-SIO module proceeding to the right. The first analog input is mapped in `Register 5y60`.
Each analog input occupies one register storing the analog value, and one register storing the configuration.
Temperature terminals are treated like inputs.

Numbering the analog outputs

Analog outputs are numbered consecutively beginning from the JX-SIO module, then proceeding to the right. The first analog output is mapped in `Register 6y60`.
Each analog output occupies one register storing the analog value, and one register storing the configuration.

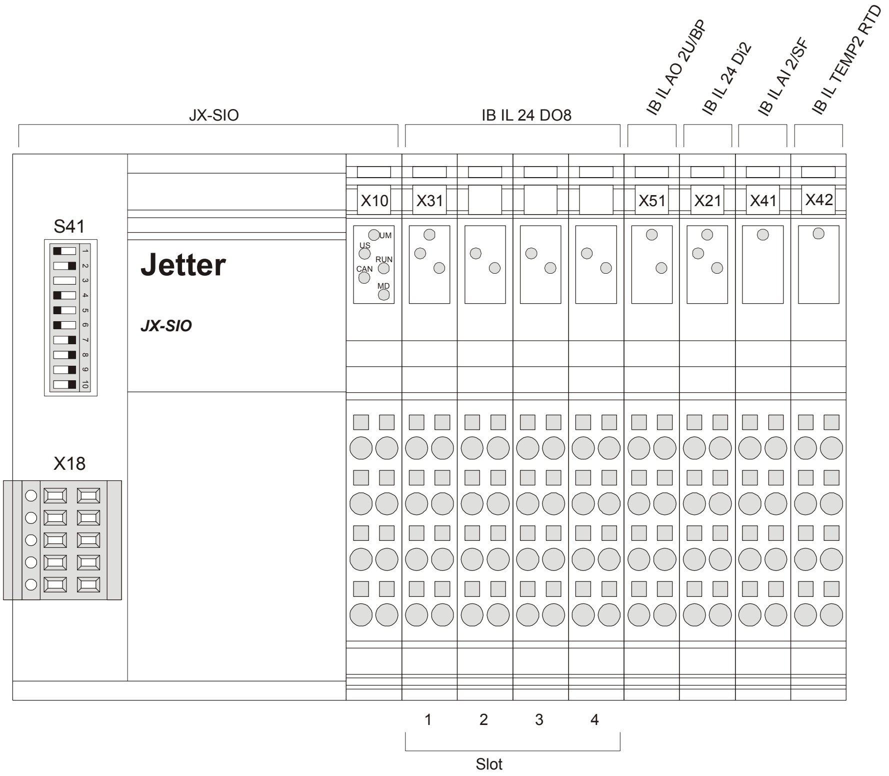
3.2 Addressing the Digital Inputs and Outputs

The address is made up of the module number and the number of the respective input or output.



Digital inputs and outputs are counted consecutively beginning from the JX-SIO module proceeding to the right. To the first digital input or output the number `7y01` assigned. For numbering purposes, analog inputs, analog outputs and temperature terminals are being ignored. The highest number of a digital input is `IN 7y64`, and of a digital output it is `OUT 7y64`.

Example: Addressing the Digital Inputs and Outputs



Configuration of a JX-SIO Module

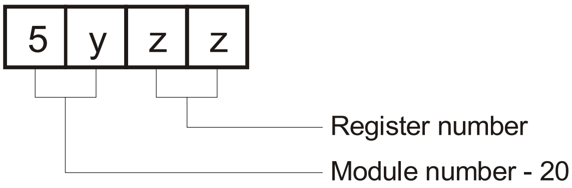
A JX-SIO module with the a.m. configuration has got the following input and output numbers given in the table below:

Analog Input Numbers		
Terminal	**IO Number**	**Description**
X31 IB IL 24 DO 8	`OUT 7y01`	Output slot 1, clamping point 1.1
	`OUT 7y02`	Output slot 1, Clamping point 2.1
	`OUT 7y03`	Output slot 2, Clamping point 1.1
	`OUT 7y04`	Output slot 2, Clamping point 2.1
	`OUT 7y05`	Output slot 3, Clamping point 1.1
	`OUT 7y06`	Output slot 3, Clamping point 2.1
	`OUT 7y07`	Output slot 4, Clamping point 1.1
	`OUT 7y08`	Output slot 4, Clamping point 2.1
X21 IB IL 24 DI 2	`IN 7y01`	Input, clamping point 1.1
	`IN 7y02`	Input, clamping point 2.1

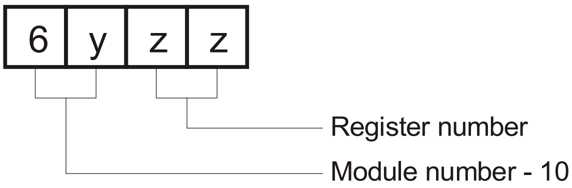
3.3 Access to IOs through Overlaying of Registers

Overlaying of registers makes for simultaneous reading of several inputs by using only one SYMPAS instruction. Furthermore, several outputs can be read out or written by using only one SYMPAS instruction.

The register address for inputs is made up of the module number minus 20 and the number of the respective overlaid register.



The register address for outputs is made up of the module number minus 10 and the number of the respective overlaid register.



Example: Resetting all outputs

This samples program is used to reset all 8 outputs located on terminal IB IL 24 DO 8 shown in the sample configuration. The JX-SIO has got the module number 70.

Program

```
REGZERO 6010                   // resetting the outputs 7001...7016
```

<table>
<tr><th colspan="8">Register 6y00: Overlaid Outputs 7y01…7y24</th></tr>
<tr><td>Bit 23</td><td>Bit 22</td><td>Bit 21</td><td>Bit 20</td><td>Bit 19</td><td>Bit 18</td><td>Bit 17</td><td>Bit 16</td></tr>
<tr><td>7y24</td><td>7y23</td><td>7y22</td><td>7y21</td><td>7y20</td><td>7y19</td><td>7y18</td><td>7y17</td></tr>
<tr><td>Bit 15</td><td>Bit 14</td><td>Bit 13</td><td>Bit 12</td><td>Bit 11</td><td>Bit 10</td><td>Bit 9</td><td>Bit 8</td></tr>
<tr><td>7y16</td><td>7y15</td><td>7y14</td><td>7y13</td><td>7y12</td><td>7y11</td><td>7y10</td><td>7y09</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>7y08</td><td>7y07</td><td>7y06</td><td>7y05</td><td>7y04</td><td>7y03</td><td>7y02</td><td>7y01</td></tr>
</table>

<table>
<tr><th colspan="8">Register 6y10: Overlaid Outputs 7y01…7y16</th></tr>
<tr><td>Bit 23</td><td>Bit 22</td><td>Bit 21</td><td>Bit 20</td><td>Bit 19</td><td>Bit 18</td><td>Bit 17</td><td>Bit 16</td></tr>
<tr><td colspan="8">Unused</td></tr>
<tr><td>Bit 15</td><td>Bit 14</td><td>Bit 13</td><td>Bit 12</td><td>Bit 11</td><td>Bit 10</td><td>Bit 9</td><td>Bit 8</td></tr>
<tr><td>7y16</td><td>7y15</td><td>7y14</td><td>7y13</td><td>7y12</td><td>7y11</td><td>7y10</td><td>7y09</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>7y08</td><td>7y07</td><td>7y06</td><td>7y05</td><td>7y04</td><td>7y03</td><td>7y02</td><td>7y01</td></tr>
</table>

<table>
<tr><th colspan="8">Register 6y20: Overlaid Outputs 7y01…7y08</th></tr>
<tr><td>Bit 23</td><td>Bit 22</td><td>Bit 21</td><td>Bit 20</td><td>Bit 19</td><td>Bit 18</td><td>Bit 17</td><td>Bit 16</td></tr>
<tr><td colspan="8">Unused</td></tr>
<tr><td>Bit 15</td><td>Bit 14</td><td>Bit 13</td><td>Bit 12</td><td>Bit 11</td><td>Bit 10</td><td>Bit 9</td><td>Bit 8</td></tr>
<tr><td colspan="8">Unused</td></tr>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>7y08</td><td>7y07</td><td>7y06</td><td>7y05</td><td>7y04</td><td>7y03</td><td>7y02</td><td>7y01</td></tr>
</table>

The examples of registers 6y00, 6y10 and 6y20 show how individual outputs are assigned to the register bits when overlaying registers with IOs. Overlaying of registers with IOs makes parallel access to inputs and outputs easier.

Register Overlaying - Inputs

32 bit register overlaying with JetControl

Register	Inputs
5x00	7y01 ... 7y32
5y01	7y09 ... 7y40
5y02	7y17 ... 7y48
5y03	7y25 ... 7y56
5y04	7y33 ... 7y64

24 bit register overlaying with NANO

Register	Inputs
5x00	7y01 ... 7y24
5y01	7y09 ... 7y32
5y02	7y17 ... 7y40
5y03	7y25 ... 7y48
5y04	7y33 ... 7y56
5y05	7y41 ... 7y64

16 bit register overlaying

Register	Inputs
5y10	7y01 ... 7y16
5y11	7y09 ... 7y24
5y12	7y17 ... 7y32
5y13	7y25 ... 7y40
5y14	7y33 ... 7y48
5y15	7y41 ... 7y56
5y16	7y49 ... 7y64

8 bit register overlaying

Register	Inputs
5y20	7y01 ... 7y08
5y21	7y09 ... 7y16
5y22	7y17 ... 7y24
5y23	7y25 ... 7y32
5y24	7y33 ... 7y40
5y25	7y41 ... 7y48
5y26	7y49 ... 7y56
5y27	7y57 ... 7y64

Register Overlaying - Outputs

32 bit register overlaying with JetControl

Register	Outputs
6y00	7y01 ... 7y32
6y01	7y09 ... 7y40
6y02	7y17 ... 7y48
6y03	7y25 ... 7y56
6y04	7y33 ... 7y64

24 bit register overlaying with NANO

Register	Outputs
6y00	7y01 ... 7y24
6y01	7y09 ... 7y32
6y02	7y17 ... 7y40
6y03	7y25 ... 7y48
6y04	7y33 ... 7y56
6y05	7y41 ... 7y64

16 bit register overlaying

Register	Outputs
6y10	7y01 ... 7y16
6y11	7y09 ... 7y24
6y12	7y17 ... 7y32
6y13	7y25 ... 7y40
6y14	7y33 ... 7y48
6y15	7y41 ... 7y56
6y16	7y49 ... 7y64

8 bit register overlaying

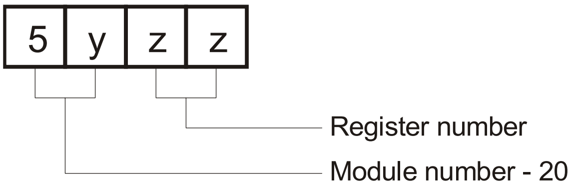
Register	Outputs
6y20	7y01 ... 7y08
6y21	7y09 ... 7y16
6y22	7y17 ... 7y24
6y23	7y25 ... 7y32
6y24	7y33 ... 7y40
6y25	7y41 ... 7y48
6y26	7y49 ... 7y56
6y27	7y57 ... 7y64

3.4 Registers for Analog Inputs

3.4.1 Reading Analog Input Values

Each analog input can be read through a register. Temperature terminals are treated like inputs.

The register address for analog inputs is made up of the module number minus 20 and the number of the corresponding register:



Registers for Analog Inputs	
Register	**Inputs**
5y60	1st analog input
5y61	2nd analog input
5y62	3rd analog input
5y63	4th analog input
5y64	5th analog input
5y65	6th analog input
5y66	7th analog input
5y67	8th analog input
5y68	9th analog input
5y69	10th analog input
5y70	11th analog input
5y71	12th analog input

3.4.2 Value Range of analog inputs

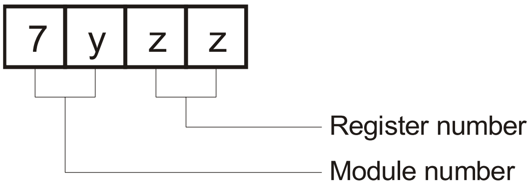
The values of anlog inputs can be displayed with or without sign. Because analog input values have always a size of 16 bits the range without sign will be 0 ... 65535 and without sign will be–32768 ... +32767.

Register 7y09: Value Range of analog inputs	
Funktion	**Beschreibung**
Read	current value ranges, this register is bit-oriented
Write	new value ranges Bit 0 : 0 = 1st analog input 0 ... 65535 1 = 1st analog input –32768 ... +32767 Bit 1 : 0 = 2nd analog input 0 ... 65535 1 = 2nd analog input –32768 ... +32767 Bit 2 : 0 = 3rd analog input 0 ... 65535 1 = 3rd analog input –32768 ... +32767 etc.
Value Range	0 – 4095
Reset Value	0

3.4.3 Configuring Analog Inputs

Each analog input can be configured through a register. Temperature terminals are treated like inputs.
The data sheet belonging to the corresponding terminal gives a detailed description on how to configure an analog input terminal. The process data output words specified in the data sheet can be transferred to the terminals through the registers 7y10 to 7y21.

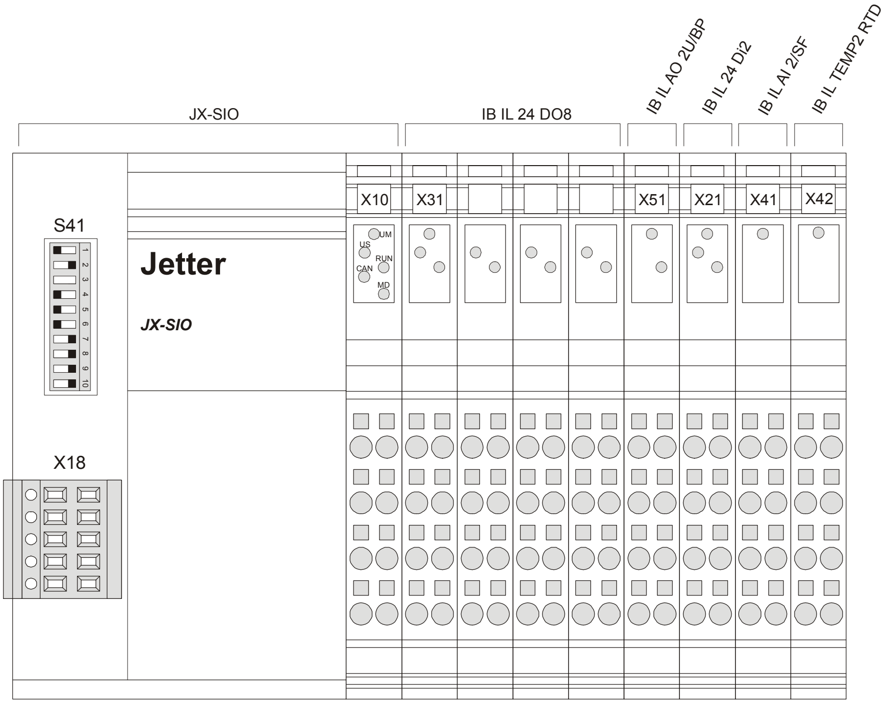
The register number for configuration of analog inputs is made up of the module number and the number of the corresponding register:



Registers for Configuration of Analog Inputs

Register	Inputs
7y10	Configuration of or process data output word for the 1st analog input
7y11	Configuration of or process data output word for the 2nd analog input
7y12	Configuration of or process data output word for the 3rd analog input
7y13	Configuration of or process data output word for the 4th analog input
7y14	Configuration of or process data output word for the 5th analog input
7y15	Configuration of or process data output word for the 6th analog input
7y16	Configuration of or process data output word for the 7th analog input
7y17	Configuration of or process data output word for the 8th analog input
7y18	Configuration of or process data output word for the 9th analog input
7y19	Configuration of or process data output word for the 10th analog input
7y20	Configuration of or process data output word for the 11th analog input
7y21	Configuration of or process data output word for the 12th analog input

Example: Addressing Analog Inputs



Configuration of a JX-SIO Modules

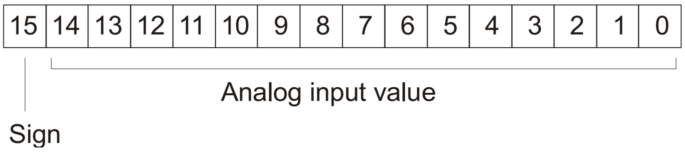
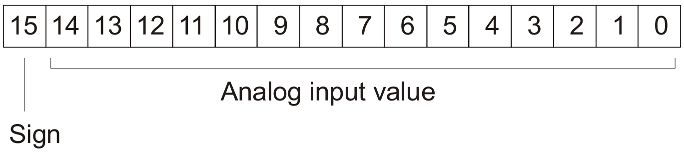
The analog inputs and their configuration registers on the JX-SIO module with the shown configuration are assigned to the following register numbers:

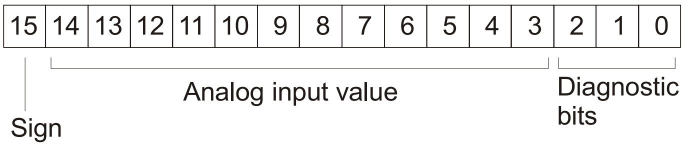
Analog Input Numbers		
Terminal	**Register**	**Description**
X41 IB IL AI 2/SF	5y60	Input value of the first analog input channel
	5y61	Input value of the second analog input channel
	7y10	Process data output word OUT[0] for setting the filter, the format and the measuring range for the first analog input channel
	7y11	Process data output word OUT[1] for setting the filter, the format and the measuring range of the IB IL AI 2/SF (X42) module for the second analog input channel
X42 IB IL TEMP 2 RTD	5y62	Input value of the first temperature input channel
	5y63	Input value of the second temperature input channel
	7y12	Process data input word for configuring the first temperature input channel
	7y13	Process data input word for configuring the second temperature input channel

3.5 Analog Input Terminals - Overview

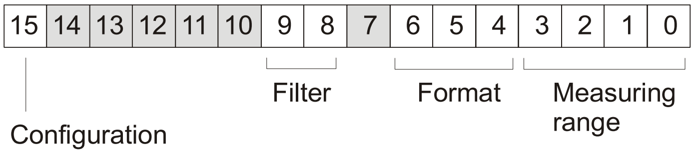
3.5.1 Analog Input Terminal Module IB IL AI 2/SF

This manual describing the analog input terminal module IB IL AI 2/SF is intended to be used in conjunction with the data sheet # 5564A of Phoenix Contact GmbH & Co.

Register 5y60 ... 5y71: IB IL AI 2/SF - Analog Input Value	
IL Format and Standardized Representation	
Function	**Description**
Read	Present analog input value, 15 bit with sign  **Extended diagnostic functions:** -32767 : Measuring range exceeded -32766 : Cable breakage -32764 : Invalid measured value, no valid measured value available -32752 : Invalid configuration -32704 : Terminal defective -32640 : Measuring range exceeded (below limit)
Write	Illegal
Value Range	-32768 ... +32767
RT Format	
Function	**Description**
Read	Present analog input value, 15 bit with sign  32767 : Cable breakage
Write	Illegal
Value Range	-32768 ... +32767

<table>
<tr><th colspan="2">ST Format
IN [1] with instruction codes 0x00$_{hex}$ and 5x00$_{hex}$</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present analog input value, 12 bit with sign and 3 diagnostic bits

Diagnostic bits:
Bit 0 = 1 : Out-of-range signal
Bit 1 = 1 : Cable breakage
Bit 2 = 1 : Measuring range from 4 to 20mA</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
</table>

Register 7y10 ... 7y21:
IB IL AI 2/SF - Configuration Value

Function	Description
Read	Present configuration
Write	New configuration  **Configuration** 0 = Default selection 1 = Configuration data are valid **Filter** 00 = 16-fold mean value (default) 01 = No filter **Format** 00 = IB IL (15 Bit) (default) 01 = IB ST (12 Bit) 10 = IB RT (15 Bit) 11 = Standardized representation **Measuring range** 0000 = 0 V to 10 V (default) 0001 = ±10 V 1000 = 0 mA to 20 mA 1001 = ±20 mA 1010 = 4 mA to 20 mA
Value Range	-32768 ... +32767
Value following reset	0

Example: Configuring an IB IL AI 2/SF module to cover a measuring range from –10 V to +10 V

In this example an analog input terminal module is configured so that it covers a measuring range from –10 V to +10 V. For this purpose, bit 15 and bit 0 of the configuration register have to be set. The IB IL AI 2/SF is the first analog input terminal module of the JX-SIO module which has got module number 70.
For displaying negative values the value range of the anlog inputs must be configured via register 7009 also.
To make the example easier, the bits are loaded into a free application register first. Upon completion of all BIT_SET instructions the terminal is to be reconfigured and the new configuration value is to be assigned to it.
It is not advisable to directly use the BIT_SET instruction for the configuration register. Each of the large number of BIT_SET instructions would cause the terminal module to be reconfigured.

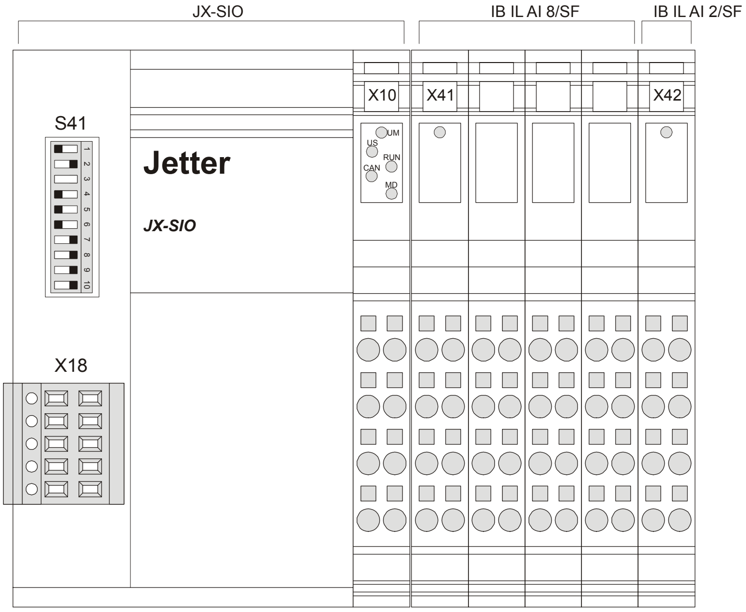
Program

```
REGISTER_LOAD (7009, 0x0003)    // configure value ranges
                                // channel 1-2: -32768 ... +32767
REGZERO 1000                    // Set application register to zero
BIT_SET (1000, 0)               // Measuring range -10 V .. +10 V
BIT_SET (1000, 15)              // Enable configuration
REGISTER_LOAD (7010, @1000)     // Write the new configuration value
                                // into the terminal module
```

3.5.2 Analog Input Terminal Module IB IL AI 8/SF

This manual describing the analog input terminal module IB IL AI 8/SF is intended to be used in conjunction with the data sheet # 6226B of Phoenix Contact GmbH & Co.
Up to 8 analog inputs can be connected to the IB IL AI8/SF module. However, these 8 analog inputs cannot be read directly. In order to read an analog input it has to be selected first. The terminal module occupies only two input registers and two configuration registers. By means ot these configuration registers 8 analog inputs can be configured and 1 analog input channel can be selected to enable read access. In the first analog input register the instruction is returned, and in the second analog input register the input value of the selected analog input channel is contained.
To read out more than one analog input channel simultaneously, group instructions can be used.

Example: Addressing Analog Inputs using the IB IL AI 8/SF module



Configuration of a JX-SIO Modules

Analog Input Numbers		
Terminal	**Register**	**Description**
X41 IB IL AI 8/SF	5y60	Process data input word IN[0]
	5y61	Process data input word IN[1]
	7y10	Process data output word OUT[0] to transfer an instruction code and select the required analog input
	7y11	Process data output word OUT[1] for setting the filter, the format and the measuring range

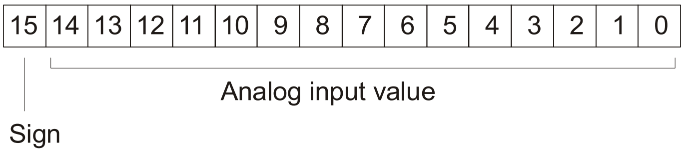
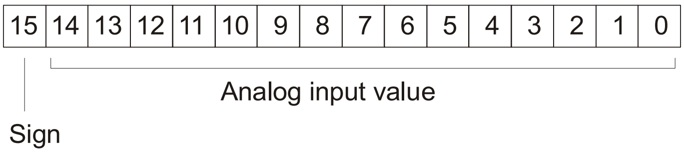
In this sample configuration of a JX-SIO, instruction codes and format settings can be transferred to the analog input terminal module IB IL AI 8/SF via register 7y10 and 7y11. Instruction codes and format settings are used to specify what is read back in registers 5y60 and 5y61.
With differing JX-SIO configurations the register numbers will change accordingly.

The register numbers refer to the exemplary configuration

Influence of Instruction Codes
Register Numbers according to Exemplary Configuration

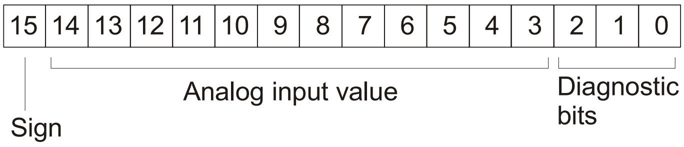
Register 7y60 Instruction Code OUT[0]	Register 5y60 IN[0]	Register 5y61 IN[1]
0x00$_{hex}$ and 5x00$_{hex}$	Instruction code mirroring	Analog input value of the selected channel
3C00$_{hex}$	Instruction code mirroring	Firmware version and module ID
1x00$_{hex}$, 4x00$_{hex}$, 6000$_{hex}$	Instruction code mirroring	Mirror image of the set configuration
7000$_{hex}$	8 bit analog input value - channel 1 of the terminal module	8 bit analog input value - channel 3 of the terminal module
	8 bit analog input value - channel 2 of the terminal module	8 bit analog input value - channel 4 of the terminal module
7100$_{hex}$	8 bit analog input value - channel 5 of the terminal module	8 bit analog input value - channel 7 of the terminal module
	8 bit analog input value - channel 6 of the terminal module	8 bit analog input value - channel 8 of the terminal module
7400$_{hex}$	16 bit analog input value - channel 1 of the terminal module	16 bit analog input value - channel 2 of the terminal module
7500$_{hex}$	16 bit analog input value - channel 3 of the terminal module	16 bit analog input value - channel 4 of the terminal module
7600$_{hex}$	16 bit analog input value - channel 5 of the terminal module	16 bit analog input value - channel 6 of the terminal module
7700$_{hex}$	16 bit analog input value - channel 7 of the terminal module	16 bit analog input value - channel 8 of the terminal module

In the above table „x“ is used to designate the respective channel number of the analog input. The analog input channel 1 is selected by entering x=0, the analog channel 2 by entering x=1.

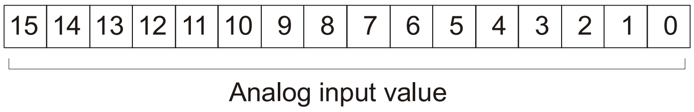
<table>
<tr><th colspan="2">Register 5y60 ... 5y71:
IB IL AI 8/SF - Analog Input Value</th></tr>
<tr><th colspan="2">IL Format and Standardized Representation
IN [1] with instruction codes 0x00_{hex} and 5x00_{hex}</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present analog input value, 15 bit with sign

Extended diagnostic functions:
-32767 : Measuring range exceeded
-32766 : Cable breakage
-32764 : Invalid measured value, no valid measured value available
-32752 : Invalid configuration
-32736 : No supply voltage for peripheral devices
-32704 : Terminal defective
-32640 : Measuring range exceeded (below limit)</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
<tr><th colspan="2">RT Format
IN [1] with instruction codes 0x00_{hex} and 5x00_{hex}</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present analog input value, 15 bit with sign

32767 : Cable breakage</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
</table>

Register 5y60 ... 5y71:
IB IL AI 8/SF - Analog Input Value

ST Format
IN [1] with instruction codes 0x00$_{\text{hex}}$ and 5x00$_{\text{hex}}$

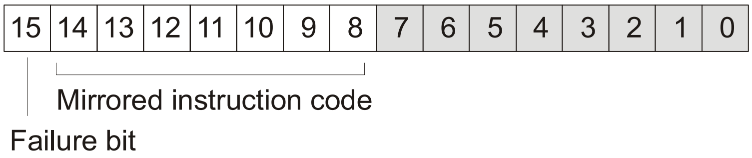
Function	Description
Read	Present analog input value, 12 bit with sign and 3 diagnostic bits  **Diagnostic bits:** Bit 0 = 1 : Out-of-range signal Bit 1 = 1 : Cable breakage Bit 2 = 1 : Measuring range from 4 to 20mA
Write	Illegal
Value Range	-32768 ... +32767

PIO Format – Current measuring range 4 mA to 20 mA only
IN [1] with instruction codes 0x00$_{\text{hex}}$ and 5x00$_{\text{hex}}$

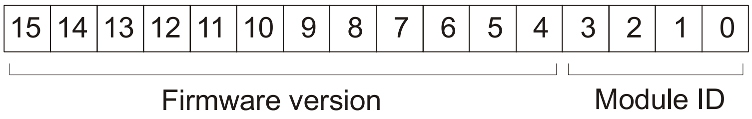
Function	Description
Read	Present analog input value, 16 bit  Despite the fact that this format is intended only for the range between 4 mA and 20 mA, signals ranging from 0 mA to 24 mA can be measured. Thus, overranges can be defined in the application program.
Write	Illegal
Value Range	-32768 ... +32767

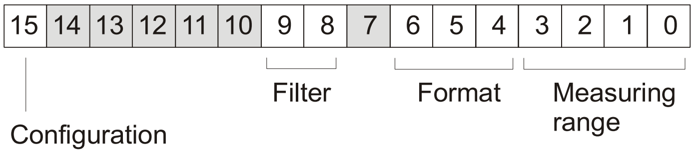
Register 5y60 ... 5y71: Additional Contents

Reading Back the Instruction Code IN[0] only with instruction codes 0x00 $_{hex}$, 5x00$_{hex}$, 3C00$_{hex}$, 1x00$_{hex}$, 4x00$_{hex}$ and 6000$_{hex}$

Function	Description
Read	Read back the preset instruction code  **Failure bit** 1 = Failure during instruction transfer **Mirrored instruction code** When the instruction code has been executed completely, it is mapped in this register
Write	Illegal
Value Range	-32768 ... +32767

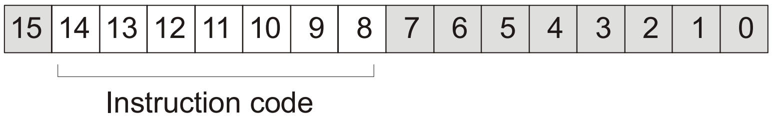
Firmware Version and Module ID IN[1] with instruction code 3C00$_{hex}$

Function	Description
Read	Firmware version and module ID  **Firmware version** For example, the value 123$_{hex}$ represents firmware version V1.23 **Module ID** 6 = IB IL AI 8/SF
Write	Illegal
Value Range	-32768 ... +32767

<table>
<tr><th colspan="2">Register 5y60 ... 5y71:
Additional Contents</th></tr>
<tr><th colspan="2">Mapping the Preset Configuration
IN[1] only with instruction code 1x00$_{hex}$</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Reading back the preset configuration
 </td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
</table>

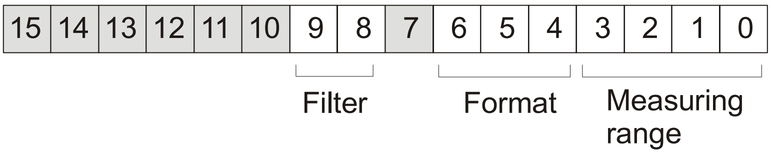
Register 7y10 ... 7y21:
IB IL AI 8/SF - Configuration Value

Process Data Output Word OUT[0] Register 7yzz

Function	Description
Read	Present instruction
Write	New instruction  **Instruction code** 0000 $Z_2 Z_1 Z_0$ = Read analog input value of channel Z-1 0010 $Z_2 Z_1 Z_0$ = Read configuration of channel Z-1 0111100 = Read firmware version and module ID 1000 $Z_2 Z_1 Z_0$ = Configure channel Z-1 1010 $Z_2 Z_1 Z_0$ = Configure channel Z-1 and read analog input value of channel Z-1 1100000 = Configure complete terminal module 1110000 = Read analog input value of channel 1 … 4 1110001 = Read analog input value of channel 5 … 8 1110100 = Read analog input value of channel 1 … 2 1110101 = Read analog input value of channel 3 … 4 1110110 = Read analog input value of channel 5 … 6 1110111 = Read analog input value of channel 7 … 8
Value Range	-32768 ... +32767

Register 7y10 ... 7y21:
IB IL AI 8/SF - Configuration Value

Process Data Output Word OUT[0] Register 7yzz + 1

Function	Description
Read	Present configuration
Write	New configuration  **Filter** 00 = 16-fold mean value (default) 01 = No filter 10 = 4-fold mean value 11 = 32-fold mean value **Format** 000 = IB IL (15 Bit) (default) 001 = IB ST (12 bit) 010 = IB RT (15 bit) 011 = Standardized representation 100 = PIO only for the range from 4 mA to 20 mA **Measuring range** 0000 = 0 V to 10 V (default) 0001 = ±10 V 0010 = 0 V through 5 V
Value Range	-32768 ... +32767

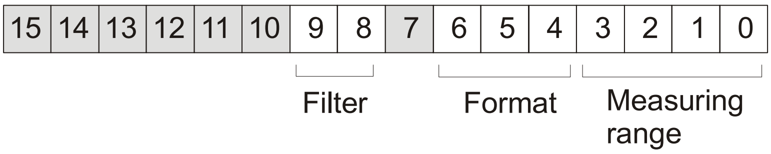
The two configuration registers of the IB IL AI 8/SF module occupy two consecutive registers.

Example: IB IL AI 8/SF Configuration

In this example, individual channels of the analog input terminal module IB IL AI 8/SF are configured differently. Then, analog input values are cyclically read in from the inidividual channels and stored to registers 1000 through 1003.
The JX-SIO module number is 70 and the analog input module IB IL AI 8/SF is the first module plugged into the rack.

Configuration OUT[1]			
Channel #	**Filter**	**Format**	**Measuring range**
1	16-fold = 00	IB IL = 000	0 V ... 10 V = 0000
2	none = 01	IB IL = 000	± 10 V = 0001
3	4-fold = 10	IB ST = 001	0 V ... 25 V = 0100
4	32-fold = 11	PIO = 100	4 mA 20 mA = 1010

First of all, on the basis of the given filter, format and measuring range settings the required configuration parameters have to be calculated. The configuration parameters have to be written into register 7011 after the corresponding channel has been selected by means of register 7010.



Configuration data for individual channels are calculated according to the table shown above. For example, configuration datum for channel 2 in hexadecimal representation is 0101_{hex}.

Program

```
REGISTER_LOAD (7009, 0x000F)        // Configuration value ranges
                                    // channel # 1-4: -32768 ... +32767
REGISTER_LOAD (7010, 0x4000)        // Configuration - channel # 1
WHEN
    REG 7010 = REG 5060             // Instruction acknowledged
THEN
    REGISTER_LOAD (7011, 0x0000)    // Configuration data - channel # 1

REGISTER_LOAD (7010, 0x4100)        // Configuration - channel # 2
WHEN
    REG 7010 = REG 5060             // Instruction acknowledged
THEN
    REGISTER_LOAD (7011, 0x0101)    // Configuration data - channel # 2

REGISTER_LOAD (7010, 0x4200)        // Configuration - channel # 3
WHEN
    REG 7010 = REG 5060             // Instruction acknowledged
```

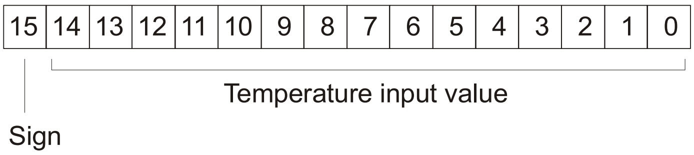
```
THEN
    REGISTER_LOAD (7011, 0x0214)    // Configuration data - channel # 3

REGISTER_LOAD (7010, 0x4300)        // Configuration - channel # 4
WHEN
    REG 7010 = REG 5060             // Instruction acknowledged
THEN
    REGISTER_LOAD (7011, 0x034A)    // Configuration data - channel # 4

LABEL sReadLoop
REGISTER_LOAD (7010, 0x0000)        // Select channel # 1
WHEN
    REG 7010 = REG 5060             // Instruction acknowledged
THEN
    REGISTER_LOAD (1000, @5061)     // Store input value to REG 1000

REGISTER_LOAD (7010, 0x0100)        // Select channel # 2
WHEN
    REG 7010 = REG 5060             // Instruction acknowledged
THEN
    REGISTER_LOAD (1001, @5061)     // Store input value to REG 1001
REGISTER_LOAD (7010, 0x0200)        // Select channel # 3
WHEN
    REG 7010 = REG 5060             // Instruction acknowledged
THEN
    REGISTER_LOAD (1002, @5061)     // Store input value to REG 1002
REGISTER_LOAD (7010, 0x0300)        // Select channel # 4
WHEN
    REG 7010 = REG 5060             // Instruction acknowledged
THEN
    REGISTER_LOAD (1003, @5061)     // Store input value to REG 1003
    GOTO sReadLoop                  // and return to the start
```

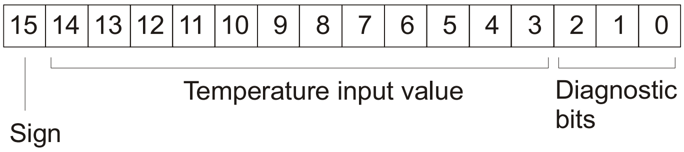
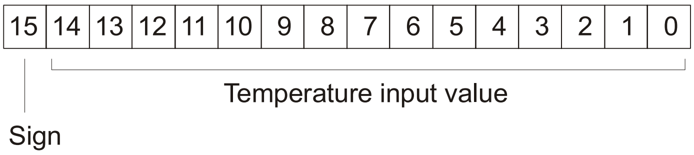
3.5.3 Temperature Terminal Module IB IL TEMP 2 RTD

This manual describing the temperature terminal module IB IL TEMP 2/RTD is intended to be used in conjunction with the data sheet # 5755B of Phoenix Contact GmbH & Co.

Register 5y60 ... 5y71:
Temperature Input Value IB IL TEMP 2 RTD

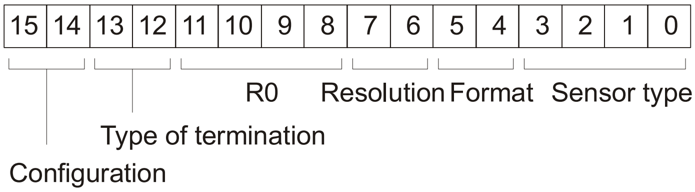
Format 1

Function	Description
Read	Present temperature input value, 15 bit with sign  **Extended diagnostic functions:** -32767 : Measuring range exceeded -32766 : Wire breakage or short-circuit (available only in the temperature range) -32764 : Invalid measured value, no valid measured value available -32752 : Invalid configuration -32704 : Terminal defective -32640 : Measuring range exceeded (below limit)
Write	Illegal
Value Range	-32768 ... +32767
Value following reset	Value present at the temperature value

<table>
<tr><th colspan="2">Register 5y60 ... 5y71:
Temperature Input Value IB IL TEMP 2 RTD</th></tr>
<tr><th colspan="2">Format 2</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present temperature input value, 12 bit with sign and 3 diagnostic bits

Diagnostic bits:
Bit 0 = 1 : Out-of-range signal
Bit 1 = 1 : Cable breakage
Bit 2 : Reserved</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
<tr><td>Value following reset</td><td>Value present at the temperature value</td></tr>
<tr><th colspan="2">Format 3</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present temperature input value, 15 bit with sign

-32766 : Cable breakage</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
<tr><td>Value following reset</td><td>Value present at the temperature value</td></tr>
</table>

Register 7y10 ... 7y21:
IB IL TEMP 2 RTD - Configuration Value

Function	Description
Read	Present configuration
Write	New configuration



Configuration

00 = Default selection

10 = Configuration data are valid

Type of termination

00 = 3-wire termination

01 = 2-wire termination

10 = 4-wire termination (only channel # 1)

R0

0000	=	100Ω	1000	=	240Ω
0001	=	10Ω	1001	=	300Ω
0010	=	20Ω	1010	=	400Ω
0011	=	30Ω	1011	=	500Ω
0100	=	50Ω	1100	=	1000Ω
0101	=	120Ω	1101	=	1500Ω
0110	=	150Ω	1110	=	2000Ω
0111	=	200Ω	1111	=	3000Ω (adjustable)

Resolution for sensor types 0000 through 1010

00 = 0.1 °C

01 = 0.01 °C

10 = 0.1 °F

11 = 0.01 °F

Resolution for sensor type 1101

00 = 1%

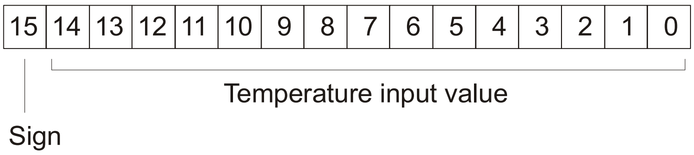
01 = 0,1%

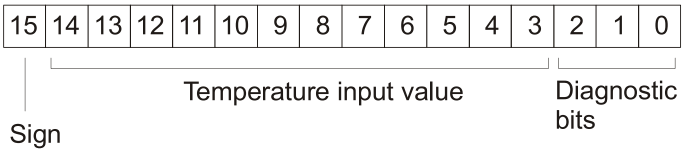
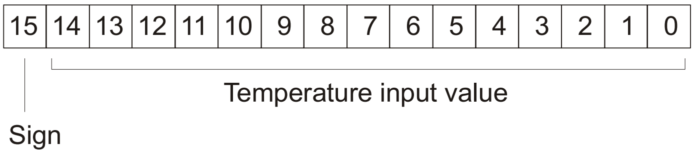
Register 7y10 ... 7y21:
IB IL TEMP 2 RTD - Configuration Value

	Resolution for sensor type 1110 00 = 0,1Ω 01 = 0,01Ω **Resolution for sensor type 1111** 00 = 1,0Ω 01 = 0,1Ω **Format** 00 = Format 1, IB standard (default) 15 bit + sign + extended diagnostics 01 = Format 2 12 bit + sign + 3 diagnostic bits 10 = Format 3 15 bit + sign **Sensor type** 0000 = Pt DIN 0001 = Pt SAMA 0010 = Ni DIN 0011 = Ni SAMA 0100 = Cu 10 0101 = Cu 50 0110 = Cu 53 0111 = Ni 1000 Landis & Gyr 1000 = Ni 500 Viessmann 1001 = KTY 81-100 1010 = KTY 84 1101 = Potentiometer [%] 1110 = linear 0 Ω through 400 Ω 1111 = linear 0 Ω through 4000 Ω
Value Range	-32768 ... +32767
Value following reset	0

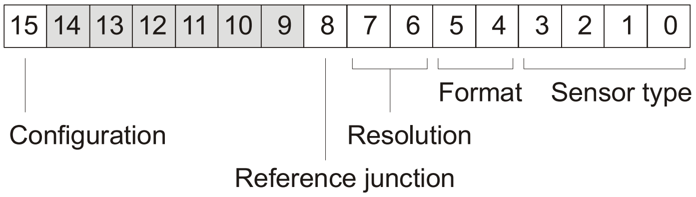
3.5.4 Temperature Terminal Module IB IL TEMP 2 UTH

This manual describing the temperature terminal module IB IL TEMP 2 UTH is intended to be used in conjunction with the data sheet # 5722B of Phoenix Contact GmbH & Co.

<table>
<tr><th colspan="2">Register 5y60 ... 5y71:
Temperature Input Value IB IL TEMP 2 UTH</th></tr>
<tr><th colspan="2">Format 1</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present temperature input value, 15 bit with sign

Extended diagnostic functions:
-32767 : Measuring range exceeded
-32766 : Cable breakage
-32764 : Invalid measured value, no valid measured value available
-32760 : Reference junction defective
-32752 : Invalid configuration
-32704 : Terminal defective</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
<tr><td>Value following reset</td><td>Value present at the temperature value</td></tr>
</table>

<table>
<tr><th colspan="2">Format 2</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present temperature input value, 12 bit with sign and 3 diagnostic bits

Diagnostic bits:
Bit 0 = 1 : Out-of-range signal
Bit 1 = 1 : Cable breakage
Bit 2 : Reserved</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
<tr><td>Value following reset</td><td>Value present at the temperature value</td></tr>
<tr><th colspan="2">Format 3</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present temperature input value, 15 bit with sign

-32766 : Cable breakage</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
<tr><td>Value following reset</td><td>Value present at the temperature value</td></tr>
</table>

Register 7y10 ... 7y21:
IB IL TEMP 2 UTH - Configuration Value

Function	Description
Read	Present configuration
Write	New configuration  **Configuration** 0 = Default selection 1 = Configuration data are valid **Reference junction** 0 = internal 1 = None **Resolution in relation to format 1** 00 = 0.1 °C (1 µV) 01 = 1 °C (10 µV) 10 = 0.1 °F 11 = 1 °F **Format** 00 = Format 1, IB standard (default) 15 bit + sign + extended diagnostics 01 = Format 2 12 bit + sign + 3 diagnostic bits 10 = Format 3 15 bit + sign **Sensor type** 0000 = TC type K 0001 = TC type J 0010 = TC type E 0011 = TC type R 0100 = TC type S 0101 = TC type T 0110 = TC type B 0111 = TC type N

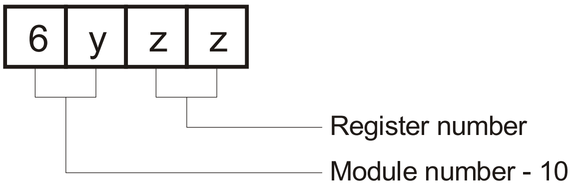
<table>
<tr><th colspan="2">Register 7y10 ... 7y21:
IB IL TEMP 2 UTH - Configuration Value</th></tr>
<tr><td></td><td>1000 = TC type U
1001 = TC type L
1010 = TC type C
1011 = TC type W
1100 = TC type HK
1101 = Reference junction (cold junction, CJ)
1110 = Voltage (-15 mV through +85 mV)</td></tr>
<tr><td>Value Range</td><td>-32768 ... +32767</td></tr>
<tr><td>Value following reset</td><td>0</td></tr>
</table>

3.6 Registers for Analog Outputs

3.6.1 Writing of Analog Output Values

Each analog output has got a size of 16 bits and can be read or written by means of a register.

The register address for analog outputs is made up of the module number minus 10 and the number of the corresponding register:



Analog Output Registers	
16-bit Analog Outputs	
Register	**Outputs**
6y60	Analog Output # 1
6y61	Analog Output # 2
6y62	Analog Output # 3
6y63	Analog Output # 4
6y64	Analog Output # 5
6y65	Analog Output # 6
6y66	Analog Output # 7
6y67	Analog Output # 8
6y68	Analog Output # 9
6y69	Analog Output # 10
6y70	Analog Output # 11
6y71	Analog Output # 12

3.6.2 Value Range of analog outputs

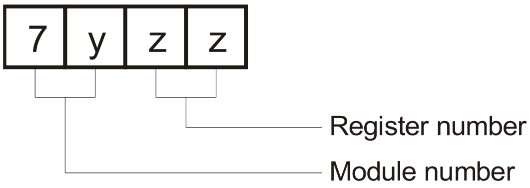
The values of anlog outputs can be displayed with or without sign. Because analog output values have always a size of 16 bits the range without sign will be 0 ... 65535 and without sign will be–32768 ... +32767.

Register 7y09: Value Range of analog inputs	
Funktion	**Beschreibung**
Read	current value ranges, this register is bit-oriented
Write	new value ranges Bit 0 : 0 = 1st analog output 0 ... 65535 1 = 1st analog output –32768 ... +32767 Bit 1 : 0 = 2nd analog output 0 ... 65535 1 = 2nd analog output –32768 ... +32767 Bit 2 : 0 = 3rd analog output 0 ... 65535 1 = 3rd analog output –32768 ... +32767 etc.
Value Range	0 – 4095
Reset Value	0

.

3.6.3 Configuring Analog Outputs

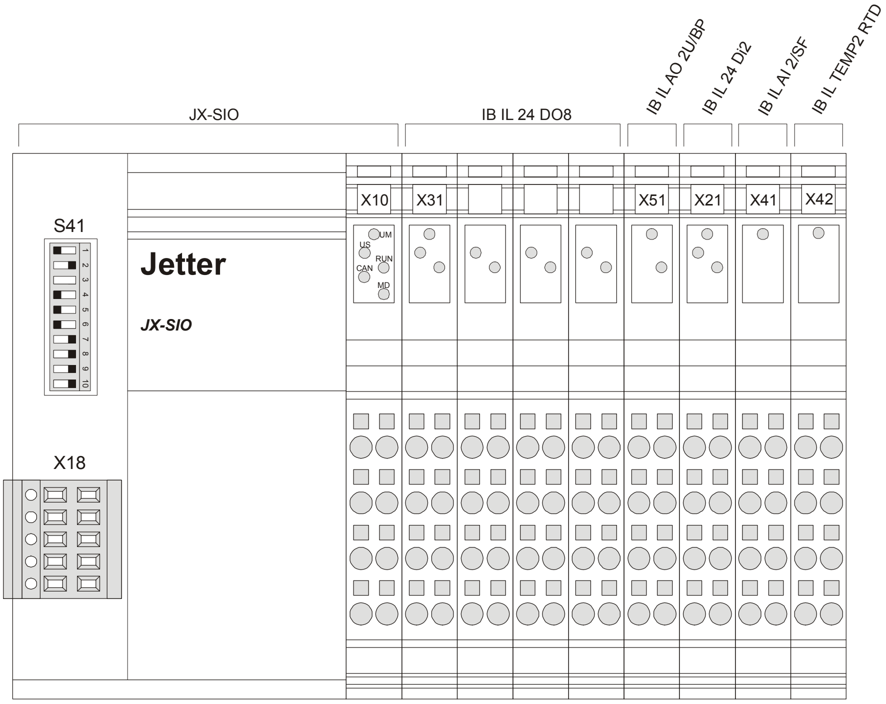
Each analog output can be configured by means of a register. The register number for configuration of analog outputs is made up of the module number and the number of the corresponding register:



Registers for the Configuration of Analog Outputs	
Register	**Inputs**
7y30	Configuration of analog output # 1
7y31	Configuration of analog output # 2
7y32	Configuration of analog output # 3
7y33	Configuration of analog output # 4
7y34	Configuration of analog output # 5
7y35	Configuration of analog output # 6
7y36	Configuration of analog output # 7
7y37	Configuration of analog output # 8
7y38	Configuration of analog output # 9
7y39	Configuration of analog output # 10
7y40	Configuration of analog output # 11
7y41	Configuration of analog output # 12

In order to configure an analog output terminal, the parameterization code has to be written into registers 7y30 through 7y41. The parameterization code has to be taken from the data sheet of the corresponding analog output terminal module.

Example: Addressing Scheme for Analog Outputs



Configuration of a JX-SIO Modules

The analog outputs on the JX-SIO module with the given configuration are assigned to the following register numbers:

Analog Output Numbering		
Terminal	**Register**	**Description**
X51 IB IL AO 2/U/BP	6y60	Output value of the first analog input
	6y61	Output value of the second analog input
	7y30	Reading back the output value of the first analog input and configuring the first analog input
	7y31	Reading back the output value of the second analog input and configuring the second analog input

3.7 Analog Output Terminals - Overview

3.7.1 Analog Output Terminal Module IB IL AO 2/U/BP

This manual describing the analog output terminal module IB IL AO 2/U/BP is intended to be used in conjunction with the data sheet # 5660A of Phoenix Contact GmbH & Co.

3.8 Response of Digital and Analog I/Os to Errors

3.8.1 Monitoring Interval

Monitoring telegrams are periodically exchanged between controller and JX-SIO module via system bus. By doing so, the control system can detect whether the connection to a JX-SIO module is interrupted. In the case of an interruption, register 2008 bit 4 “Timeout of an IO Module” is set and the number of the relevant module is entered into register 2011. Connection to the faulty JX-SIO module can be re-established only after a restart of the control system.
The interval between two monitoring telegrams can be set in register 2028.

Register 2028: Monitoring Interval of I/O Modules	
Function	**Description**
Read	Monitoring interval for I/O modules in 10 ms increments
Write	New monitoring interval
Value Range	0 – 255
Value following reset	20

The JX-SIO module evaluates the receipt of these monitoring telegrams and can respond to missing telegrams. The maximum interval between two monitoring telegrams can be set for each JX-SIO module individually.

Register 7x98: Monitoring Interval of the JX-SIO Module	
Function	**Description**
Read	Present monitoring interval of the JX-SIO module in 100 ms increments
Write	New monitoring interval of the JX-SIO Module 0 : Disabling the monitoring function
Value Range	0 – 255
Value following reset	20 (2000 ms)

3.8.2 Configuring the Response of Digital Outputs to Errors

The way a JX-SIO module responds to missing monitoring telegrams can be configured for each output individually. In principle, you can specify whether an output should maintain its state or to which value it should be set.

Register 7y75: Index for Digital Outputs	
Function	**Description**
Read	Actually selected group of 8 digital outputs
Write	Selection of a new group of 8 digital outputs 0: Amount of groups of 8 outputs each 1: Digital outputs 7y01 ... 7y08 2: Digital outputs 7y09 ... 7y16 ...
Value Range	0 – 8
Value following reset	0

Register 7y78: Failure Mode of Digital Outputs	
Function	**Description**
Read	Present failure mode of the group of 8 digital outputs selected through register 7y75. The content of this register is bit-oriented. If the value 0 is contained in register 7y75, the amount of groups of 8 outputs each is read out of this register.
Write	New error mode Bit = 1 : The output assumes the value contained in register 7y79. Bit = 0 : The output maintains its state
Value Range	0 – 255
Value following reset	255

Register 7y79: Error Condition of Digital Outputs

Function	Description
Read	Present failure condition of the group of 8 digital outputs selected through register 7y75. The content of this register is bit-oriented. If the value 0 is contained in register 7y75, the amount of groups of 8 outputs each is read out of this register.
Write	New error condition Applies only in case the corresponding bit for the given output in register 7y78 is 1. Bit = 1 : Set output Bit = 0 : Reset output
Value Range	0 – 255
Value following reset	0

Example: Configuring the Response of Digital Outputs to Errors

The given example demonstrates the following: With an interrupted connection to the controller, output 7001 is to be set, output 7002 is to be reset, and output 7003 is to maintain its state.
To do so, the first group of 8 digital outputs has to be selected in register 7y75.
Error mode for outputs 7001 and 7002 has to be set to 1, error mode for output 7003 must be 0. Error condition for output 7001 has to be set to 1, and for output 7002 to 0. Error condition for output 7003 will not be evaluated.

Program

```
REGISTER_LOAD (7075, 1)            // Select first group of 8 outputs
REGISTER_LOAD (7078, 0b00000011)   // Error mode
                                   // Output 7001 -> Bit 0 = 1
                                   // Output 7002 -> Bit 1 = 1
                                   // Output 7003 -> Bit 2 = 0
REGISTER_LOAD (7079, 0b00000001)   // Error condition
                                   // Output 7001 -> Bit 0 = 1
                                   // Output 7002 -> Bit 1 = 0
                                   // Output 7003 -> Any value
```

3.8.3 Configuring the Response of Analog Outputs to Errors

The way a JX-SIO module responds to missing monitoring telegrams can be configured for each output individually. In principle, you can specify whether an output should maintain its state or to which value it should be set.

Register 7y85: Index for Analog Outputs	
Function	**Description**
Read	Actually selected analog output
Write	Select a new analog output 0: Amount of analog outputs 1: Analog output 6y60 2: Analog output 6y61 ...
Value Range	0 – 12
Value following reset	0

Register 7y88: Error Mode of Analog Outputs	
Function	**Description**
Read	Present error mode of the analog output selected through register 7y85. If the value 0 is contained in register 7y85, the amount of analog outputs is read out of this register.
Write	New error mode 1 : The output assumes the value contained in register 7y79. 0 : The output maintains its state
Value Range	0 – 1
Value following reset	1

Register 7y89: Error Condition of Analog Outputs

Function	Description
Read	Present error condition of the analog outputs selected through register 7y75. If the value 0 is contained in register 7y85, the amount of analog outputs is read out of this register.
Write	New error condition; corresponds to the analog value of an analog output. This applies only in case 1 is contained in register 7y88.
Value Range	0 – 65535
Value following reset	0

Example: Configuring the Response of Analog Outputs to Errors

The given example demonstrates the following: With an interrupted connection to the controller, output 6060 is to output the value 0, and output 6061 is to maintain its state.
To do so, the analog outputs have to be selected in register 7y85. Then, their error response has to be configured in registers 7y88, and 7y89.

Program

```
REGISTER_LOAD (7085, 1)              // Select analog output 6060
REGISTER_LOAD (7088, 1)              // Error mode
                                     // Take over the value contained in
                                     // register 7089
REGISTER_LOAD (7089, 0)              // Error state
REGISTER_LOAD (7085, 2)              // Select analog output 6061
REGISTER_LOAD (7088, 0)              // Error mode
                                     // Maintain value
```

4 Register Description

4.1 System Bus Management

Register 2011: I/O Module Timeout	
Function	**Description**
Read	Timeout of I/O, JX-SIO and Festo CPV-Direct modules
Write	Reset
Value Range	0 – 79
Value following reset	0

If within a given period of time there is no response from the JX-SIO module, the number of the corresponding JX-SIO module is entered into this register.

Register 2013: Amount of I/O Modules	
Function	**Description**
Read	Quantity of connected non-intelligent I/O modules
Write	Illegal
Value Range	0 – 31
Value following reset	Quantity

The amount of all detected JX2-IO and JX-SIO modules, as well as Festo CPV Direct valve terminals is written into this register by the controller.
The amount of detected JX-SIO and CPV Direct modules is calculated on the basis of the module with the highes module number, and not on the basis of the actual amount. If, for example, two JX-SIO modules, to which the module numbers 70 and 74 have been assigned, are connected to the system bus, these two modules count as 5 IO modules. The absent modules, numbered by 71, 72 and 73, are regarded as dummy modules.

Register 2015: Index to Module Array	
Function	**Description**
Read	Selected module
Write	Selection of a specific module
Value Range	0 to x (amount of I/O modules)
Value following reset	0

Register 2016: Module Array		
Function	**Description**	
Read	Information on module	
	0 – 63	non-intelligent expansion modules
	64	JX-SIO
	65	Festo CPV Direct
	66	Festo Terminal CPX
	67	Buerkert valve block type 8640
	128 – 250	intelligent expansion modules
	252	JX-SIO dummy module
	253	intelligent dummy module
	254	non-intelligent dummy module
	255	not identified
Write	Illegal	
Value Range	0 through 255	
Value following reset	Amount of expansion modules	

The type of an expansion module can be determined by means of registers 2015 and 2016. If there are gaps between JX-SIO module numbers, for every absent JX-SIO module value 252 “JX-SIO dummy module” is displayed.

Register 2070: Amount of JX-SIO Modules	
Function	**Description**
Read	Amount of connected JX-SIO and Festo CPV-Direct modules
Write	Illegal
Value Range	0 – 10
Value following reset	Quantity

The amount of detected JX-SIO and Festo CPV Direct modules is entered into this register.
The amount of detected JX-SIO and CPV Direct modules is calculated on the basis of the module with the highes module number, and not on the basis of the actual amount. If, for example, two JX-SIO modules, to which the module numbers 70 and 74 have been assigned, are connected to the system bus, these two modules count as 5 IO modules. The absent modules, numbered by 71, 72 and 73, are regarded as dummy modules.

4.2 Terminal Array of the JX-SIO Module

The actual configuration of a JX-SIO module can be determined by means of the terminal array. To do so, the terminal number is selected through a pointer to the terminal array. Then, the terminal type can be read from the terminal array register. Terminals are numbered consecutively beginning from the JX-SIO module proceeding to the right.
The length code and ID code can also be taken from the data sheet belonging to the respective terminal.

Register 7y02: Pointer to Terminal Array	
Function	**Description**
Read	Actually selected terminal
Write	Selection of a new terminal
Value Range	0 – 63
Value following reset	0

Register 7y03: Terminal Array	
Function	**Description**
Read	7y06 = 0 -> 7y07 = Amount of terminals 7y06 = 1 -> 7y07 = Length code and ID code of the first terminal 7y06 = 2 -> 7y07 = Length code and ID code of the second terminal etc. *= Length code and ID code taken from the data sheet* Bit 0...7 Length code Bit 8..0.15 ID code
Write	Illegal
Value Range	0 – 65535
Value following reset	Amount of terminals connected to the JX-SIO module

Terminal Configuration Data

Digital Input Terminals			
Smart I/O terminals	**ID code**	**Length code**	**Register 7y07**
IB IL 24 DI 2	BE	C2	48834
IB IL 24 DI 2-NPN	BE	C2	48834
IB IL 24 EDI 2	BE	C2	48834
IB IL 24 EDI 2-DESINA	BE	41	48705
IB IL 24 DI 4	BE	41	48705
IB IL 24 DI 8	BE	81	48769
IB IL 24 DI 8/T2	BE	81	48769
IB IL 24 DI 16	BE	01	48641
IB IL 120 DI 1	BE	C2	48834
IB IL 230 DI 1	BE	C2	48834
Digital Output Terminal			
Smart I/O terminals	**ID code**	**Length code**	**Register 7y07**
IB IL 24 DO 2	BD	C2	48578
IB IL 24 DO 2-2A	BD	C2	48578
IB IL 24 DO 2-NPN	BD	C2	48578
IB IL 24 EDO 2	BD	41	48449
IB IL 24 DO 4	BD	41	48449
IB IL 24 DO 8	BD	81	48513
IB IL 24 DO 8-2A	BD	81	48513
IB IL DO 16	BD	01	48385
IB IL DO 1 AC	BD	C2	48578
IB IL DO 4 AC-1A	BD	41	48449
Analog Input Terminals			
Smart I/O terminals	**ID code**	**Length code**	**Register 7y07**
IB IL AI 2/SF	7F	02	32514
IB IL AI 8/SF	5F	02	32514
IB IL AI 8/IS	5F	02	32514
IB IL TEMP 2 RTD	7F	02	32514
IB IL TEMP 2 UTH	7F	02	32514

Terminal Configuration Data

Analog Output Terminal

Smart I/O terminals	ID code	Length code	Register 7y07
IB IL AO 1/SF	7D	01	32001
IB IL AO 1/U/SF	7D	01	32001
IB IL AO2/U/BP	5B	02	23298

Relay Terminals

Smart I/O terminals	ID code	Length code	Register 7y07
IB IL 24/230 DOR 1/W	BD	C2	48578
IB IL 24/230 DOR 1/W-PC	BD	C2	48578
IB IL 24/230 DOR 4/W	BD	41	48449
IB IL 24/230 DOR 4/W-PC	BD	41	48449
IB IL DOR LV-SET			

Power and Segment Terminals

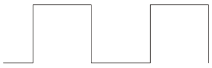
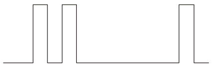
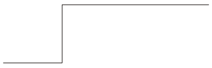
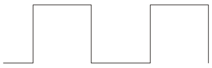
Smart I/O terminals	ID code	Length code	Register 7y07
IB IL 24 PWR IN/F-D	BE	C2	48834
IB IL 24 PWR IN/2-F-D	BE	C2	48834
IB IL 24 SEG/F-D	BE	C2	48834
IB IL 24 SEG/ELF	BE	C2	48834

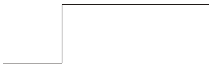
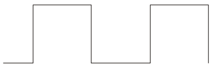
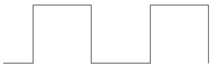
5 JX-SIO Module Diagnostics

5.1 Diagnostic Indicators (LEDs)

The JX-SIO module is equipped with 5 diagnostic LEDs.

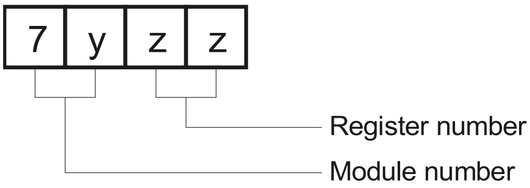
Diagnostic Indicators of the JX-SIO Module

LED	Sequence	Colour	Meaning
CAN ✸		green; flashing	The JX-SIO module has established connection to the system bus and is still initializing
CAN ✸		green; shortly flashing	The JX-SIO module is in STOP state
CAN ●		green; lit	The JX-SIO module participates in communication via system bus
CAN ✸		red; shortly flashing	Timeout of the system bus connection
CAN ●		red; lit	Fatal system bus communication error
CAN ✸		red; flashing once	Alarm condition of the system bus connection
CAN ✸		red; flashing twice	Timeout watchdog of the JX-SIO Module is active
MD ●		green; lit	Initialization of the JX-SIO module is completed
MD ✸		green / red; flashing	Self-test activated
MD ✸		red; shortly flashing	Recoverable error
MD ●		red; lit	Non-recoverable error
RUN ●		green; lit	Terminal initialization completed
RUN ✸		green; shortly flashing, 1 Hz	Peripheral fault

RUN ●		red, lit	Communication between the terminals and the JX-SIO has been stopped
CAN ✸		green / yellow; flashing	Initialization of behavior under terminal fault conditions completed
CAN ✸		red / yellow; flashing	Terminal configuration error
CAN ○		off	U_L not present
U_S ●		green; lit	U_S present
U_S ○		off	U_S not present
U_M ●		green; lit	U_M present
U_M ○		off	U_M not present

5.2 Register Addressing

The register address is made up of the module number and the respective register number:



5.3 Diagnostic Registers with NANO CPUs

<table>
<tr><th colspan="2">Register 7y90: Fault Register</th></tr>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>Read</td><td>Present value of the fault register and the most significant byte of the status register.
Fault Register
Bit 0 : Fault present
Bit 2 : Undervoltage U_M, U_S or U_L
Bit 4 : Communications fault
Bit 7 : Device-specific fault
Status Register
Bit 8 : Module address has been changed
Bit 10 : Start-up fault</td></tr>
<tr><td>Write</td><td>Illegal</td></tr>
<tr><td>Value Range</td><td></td></tr>
<tr><td>Value following reset</td><td>0 in faultless condition</td></tr>
</table>

Register 7y91: Status Register	
Function	**Description**
Read	Present value of the status register The most significant byte can be read out from register 7y90 Bit 0 ... 7 : Reserved Bit 8..0.15: Number of the first faulty terminal 0 = JX-SIO 1 = first terminal 2 = second terminal 3 = etc. Bit 16 : CRC error during communication with the terminals Bit 17 : Terminal signals peripheral fault Bit 18 : Power fault Bit 19 : Terminal configuration has been changed Bit 20 : Communication with the terminals inactive Bit 21 : Communication with the terminals inactive Bit 22 : Fault cycles
Write	Illegal
Value Range	
Value following reset	0 in faultless condition

Register 7y99: Software Version	
Function	**Description**
Read	Version of the JX-SIO software
Write	Illegal
Value Range	23 bit signed
Value following reset	Version of the JX-SIO software

5.4 Diagnostic Registers with JetControl 24x

Register 7y90: Fault Register	
Function	**Description**
Read	Present value of the fault register Bit 0: Fault present Bit 2: Voltage Bit 4: Communications fault Bit 7: Device-specific fault
Write	Illegal
Value Range	
Value following reset	0 in faultless condition

Register 7y91: Status Register	
Function	**Description**
Read	Present value of the status register Bit 0 ... 7 : Reserved Bit 8..0.15: Number of the first faulty terminal 0 = JX-SIO 1 = first terminal 2 = second terminal 3 = etc. Bit 16 : CRC error Bit 17 : Peripheral fault Bit 18 : Power fault Bit 19 : Terminal exchanged Bit 20 : Communication with the terminals inactive Bit 21 : Faulty connection Bit 22 : Fault cycles Bit 24 : Module address has been changed Bit 26 : Start-up fault
Write	Illegal
Value Range	
Value following reset	0 in faultless condition

Register 7y92: Index to Fault Array

Function	Description
Read	Present index
Write	New index
Value Range	0 – 10
Value following reset	0

Register 7y93: Fault Array

Function	Description
Read	7y92 = 0 -> 7y93 = Number of entries in the fault array 7y92 = 1 -> 7y93 = newest fault 7y92 = 2 -> 7y93 = last fault 7y92 = 3 -> 7y93 = next to last fault etc.
Write	Illegal
Value Range	31 bit signed
Value following reset	Number of entries in the fault array

Register 7y97: Serial number

Function	Description
Read	Serial number of the JX-SIO Module
Write	Illegal
Value Range	31 bit signed
Value following reset	Serial number

Register 7y99: Software Version

Function	Description
Read	Version of the JX-SIO software
Write	Illegal
Value Range	31 bit signed
Value following reset	Version of the JX-SIO software

Calculating the Extension Size

The following tables can be used to assist in calculating the extension size. First, you have to determine the IO sums of the FESTO and Smart IO modules. Then, you have to enter these sums into the corresponding lines of the system bus table. When calculating the total, you must distinguish between NANO and JetControl controllers.

IO Sums of FESTO Modules			
FESTO Module	**Quantity**	**Factor**	**Sum**
FESTO Valve Terminal CP..-GE-CO2		* 16	
FESTO Valve Terminal CP..-GE-FB		* 16	
FESTO Valve Terminal CPA		* 16	
FESTO Output Module		* 16	
FESTO Input Module		* 16	
IO Sums of FESTO Modules			

IO Sums of Smart I/O Terminals

Digital Input Terminals

Smart I/O terminals	Quantity	Factor	Sum
IB IL 24 DI 2		* 2	
IB IL 24 DI 2-NPN		* 2	
IB IL 24 EDI 2		* 2	
IB IL 24 EDI 2-DESINA		* 4	
IB IL 24 DI 4		* 4	
IB IL 24 DI 8		* 8	
IB IL 24 DI 8/T2		* 8	
IB IL 24 DI 16		* 16	
IB IL 120 DI 1		* 2	
IB IL 230 DI 1		* 2	

Digital Output Terminal

Smart I/O terminals	Quantity	Factor	Sum
IB IL 24 DO 2		* 2	
IB IL 24 DO 2-2A		* 2	
IB IL 24 DO 2-NPN		* 2	
IB IL 24 EDO 2		* 8	
IB IL 24 DO 4		* 4	
IB IL 24 DO 8		* 8	
IB IL 24 DO 8-2A		* 8	
IB IL DO 16		* 16	
IB IL DO 1 AC		* 2	
IB IL DO 4 AC-1A		* 4	

Analog Input Terminals

Smart I/O terminals	Quantity	Factor	Sum
IB IL AI 2/SF		* 4	
IB IL AI 8/SF		* 4	
IB IL AI 8/IS		* 4	
IB IL TEMP 2 RTD		* 4	
IB IL TEMP 2 UTH		* 4	

Analog Output Terminal

Smart I/O terminals	Quantity	Factor	Sum
IB IL AO 1/SF		* 2	
IB IL AO 1/U/SF		* 2	
IB IL AO2/U/BP		* 2	

<table>
<tr><th colspan="4">IO Sums of Smart I/O Terminals</th></tr>
<tr><th colspan="4">Relay Terminals</th></tr>
<tr><th>Smart I/O terminals</th><th>Quantity</th><th>Factor</th><th>Sum</th></tr>
<tr><td>IB IL 24/230 DOR 1/W</td><td></td><td>* 2</td><td></td></tr>
<tr><td>IB IL 24/230 DOR 1/W-PC</td><td></td><td>* 2</td><td></td></tr>
<tr><td>IB IL 24/230 DOR 4/W</td><td></td><td>* 4</td><td></td></tr>
<tr><td>IB IL 24/230 DOR 4/W-PC</td><td></td><td>* 4</td><td></td></tr>
<tr><td>IB IL DOR LV-SET</td><td></td><td>* 0</td><td></td></tr>
<tr><th colspan="4">Power and Segment Terminals</th></tr>
<tr><th>Smart I/O terminals</th><th>Quantity</th><th>Factor</th><th>Sum</th></tr>
<tr><td>IB IL 24 PWR IN</td><td></td><td>* 0</td><td></td></tr>
<tr><td>IB IL 24 PWR IN/F</td><td></td><td>* 0</td><td></td></tr>
<tr><td>IB IL 24 PWR IN/F-D</td><td></td><td>* 2</td><td></td></tr>
<tr><td>IB IL 24 PWR IN/2-F</td><td></td><td>* 0</td><td></td></tr>
<tr><td>IB IL 24 PWR IN/2-F-D</td><td></td><td>* 2</td><td></td></tr>
<tr><td>IB IL 24 PWR IN/R</td><td></td><td>* 0</td><td></td></tr>
<tr><td>IB IL 24 SEG</td><td></td><td>* 0</td><td></td></tr>
<tr><td>IB IL 24 SEG/F</td><td></td><td>* 0</td><td></td></tr>
<tr><td>IB IL 24 SEG/F-D</td><td></td><td>* 2</td><td></td></tr>
<tr><td>IB IL 24 SEG/ELF</td><td></td><td>* 2</td><td></td></tr>
<tr><td>IB IL 230 PWR IN</td><td></td><td>* 0</td><td></td></tr>
<tr><td>IB IL 120 PWR IN</td><td></td><td>* 0</td><td></td></tr>
<tr><td colspan="3">IO Sums of Smart I/O Terminals</td><td></td></tr>
</table>

System Bus with NANO Controller			
Module / Terminal	**Quantity**	**Factor**	**Sum**
IO Sums of FESTO Modules		* 1	
IO Sums of Smart I/O Terminals		* 1	
JX2-CNT1		* 8	
JX2-IA4		* 8	
JX2-ID8		* 8	
JX2-IO16		* 16	
JX2-OA2		* 8	
JX2-OA4		* 8	
JX2-OD2		* 8	
JX2-OD4		* 8	
JX2-OD8		* 8	
JX2-SER1		* 8	
NANO Basic Controller	1	* 16	16
Total for NANO Controller			

Maximum permissible total for NANO controllers:

Nano-B	136 digital inputs and outputs
Nano-C	136 digital inputs and outputs
Nano-D	200 digital inputs and outputs

System Bus with JetControl Controller			
Module / Terminal	**Quantity**	**Factor**	**Sum**
IO Sums of FESTO Modules		* 1	
IO Sums of Smart I/O Terminals		* 1	
JX2-CNT1		* 16	
JX2-IA4		* 16	
JX2-ID8		* 16	
JX2-IO16		* 32	
JX2-OA2		* 16	
JX2-OA4		* 16	
JX2-OD2		* 16	
JX2-OD4		* 16	
JX2-OD8		* 16	
JX2-SER1		* 16	
JetControl Basic Controller	1	* 24	24
Total for JetControl Controller			

Maximum permissible total for JetControl controllers:

JetControl 241	136 digital inputs and outputs
JetControl 243	264 digital inputs and outputs
JetControl 246	392 digital inputs and outputs

Register Overview

Register	Function	1) Value Range 2) Reset Value
32 combined digital inputs with JetControl		
5x00	Inputs 7y01 ... 7y32	1) 0 … FFFFFFFF$_{hex}$ 2) current state of inputs
5y01	Inputs 7y09 ... 7y40	
5y02	Inputs 7y17 ... 7y48	
5y03	Inputs 7y25 ... 7y56	
5y04	Inputs 7y33 ... 7y64	
24 combined digital inputs with NANO		
5x00	Inputs 7y01 ... 7y24	1) 0 … FFFFFF$_{hex}$ 2) current state of inputs
5y01	Inputs 7y09 ... 7y32	
5y02	Inputs 7y17 ... 7y40	
5y03	Inputs 7y25 ... 7y48	
5y04	Inputs 7y33 ... 7y56	
5y05	Inputs 7y41 ... 7y64	
16 combined digital inputs		
5y10	Inputs 7y01 ... 7y16	1) 0 … FFFF$_{hex}$ 2) current state of inputs
5y11	Inputs 7y09 ... 7y24	
5y12	Inputs 7y17 ... 7y32	
5y13	Inputs 7y25 ... 7y40	
5y14	Inputs 7y33 ... 7y48	
5y15	Inputs 7y41 ... 7y56	
5y16	Inputs 7y49 ... 7y64	
8 combined digital inputs		
5y20	Inputs 7y01 ... 7y08	1) 0 … FF$_{hex}$ 2) current state of inputs
5y21	Inputs 7y09 ... 7y16	
5y22	Inputs 7y17 ... 7y24	
5y23	Inputs 7y25 ... 7y32	
5y24	Inputs 7y33 ... 7y40	
5y25	Inputs 7y41 ... 7y48	
5y26	Inputs 7y49 ... 7y56	
5y27	Inputs 7y57 ... 7y64	

Register	Function	1) Value Range 2) Reset Value
32 combined digital outputs with JetControl		
6y00	Outputs 7y01 ... 7y32	1) 0 … $FFFFFFFF_{hex}$ 2) 0
6y01	Outputs 7y09 ... 7y40	
6y02	Outputs 7y17 ... 7y48	
6y03	Outputs 7y25 ... 7y56	
6y04	Outputs 7y33 ... 7y64	
24 combined digital outputs with NANO		
6y00	Outputs 7y01 ... 7y24	1) 0 … $FFFFFF_{hex}$ 2) 0
6y01	Outputs 7y09 ... 7y32	
6y02	Outputs 7y17 ... 7y40	
6y03	Outputs 7y25 ... 7y48	
6y04	Outputs 7y33 ... 7y56	
6y05	Outputs 7y41 ... 7y64	
16 combined digital outputs		
6y10	Outputs 7y01 ... 7y16	1) 0 … $FFFF_{hex}$ 2) 0
6y11	Outputs 7y09 ... 7y24	
6y12	Outputs 7y17 ... 7y32	
6y13	Outputs 7y25 ... 7y40	
6y14	Outputs 7y33 ... 7y48	
6y15	Outputs 7y41 ... 7y56	
6y16	Outputs 7y49 ... 7y64	
8 combined digital outputs		
6y20	Outputs 7y01 ... 7y08	1) 0 … FF_{hex} 2) 0
6y21	Outputs 7y09 ... 7y16	
6y22	Outputs 7y17 ... 7y24	
6y23	Outputs 7y25 ... 7y32	
6y24	Outputs 7y33 ... 7y40	
6y25	Outputs 7y41 ... 7y48	
6y26	Outputs 7y49 ... 7y56	
6y27	Outputs 7y57 ... 7y64	

Register	**Function**	**1) Value Range** **2) Reset Value**
error behavior for digital outputs		
7y75	Index for digital outputs	1) 0 … 8 2) 0
7y78	error mode for digital outputs	1) 0 … 255 2) number of entries else 255
7y79	error value for digital outputs	1) 0 - 255 2) number of entries else 255

Register	Function	1) Value Range 2) Reset Value
Analog inputs		
5y60	Analog input # 1	1) 0 … 65535 or –32768 ... +32767 depends on configured value range in register 7y09 2) value at analog input
5y61	Analog input # 2	
5y62	Analog input # 3	
5y63	Analog input # 4	
5y64	Analog input # 5	
5y65	Analog input # 6	
5y66	Analog input # 7	
5y67	Analog input # 8	
5y68	Analog input # 9	
5y69	Analog input # 10	
5y70	Analog input # 11	
5y71	Analog input # 12	
7y09	value ranges for analog inputs	1) 0 … 4095 2) 0
7y10	Configuration of analog input # 1	1) 0 … 65535 2) depends on analog input terminal
7y11	Configuration of analog input # 2	
7y12	Configuration of analog input # 3	
7y13	Configuration of analog input # 4	
7y14	Configuration of analog input # 5	
7y15	Configuration of analog input # 6	
7y16	Configuration of analog input # 7	
7y17	Configuration of analog input # 8	
7y18	Configuration of analog input # 9	
7y19	Configuration of analog input # 10	
7y20	Configuration of analog input # 11	
7y21	Configuration of analog input # 12	

Register	Function	1) Value Range 2) Reset Value
Analog outputs		
6y60	Analog output # 1	1) 0 … 65535 or –32768 ... +32767 depends on configured value range in register 7y29 2) 0
6y61	Analog output # 2	
6y62	Analog output # 3	
6y63	Analog output # 4	
6y64	Analog output # 5	
6y65	Analog output # 6	
6y66	Analog output # 7	
6y67	Analog output # 8	
6y68	Analog output # 9	
6y69	Analog output # 10	
6y70	Analog output # 11	
6y71	Analog output # 12	
7y29	value ranges for analog outputs	1) 0 … 4095 2) 0
7y30	Configuration of analog output # 1	1) 0 … 65535 2) depends on analog input terminal
7y31	Configuration of analog output # 2	
7y32	Configuration of analog output # 3	
7y33	Configuration of analog output # 4	
7y34	Configuration of analog output # 5	
7y35	Configuration of analog output # 6	
7y36	Configuration of analog output # 7	
7y37	Configuration of analog output # 8	
7y38	Configuration of analog output # 9	
7y39	Configuration of analog output # 10	
7y30	Configuration of analog output # 11	
7y31	Configuration of analog output # 12	

Register	Function	1) Value Range 2) Reset Value
error behavior for analog outputs		
7y85	Index for analog outputs	1) 0 … 12 2) 0
7y88	error mode for analog outputs	1) 0 … 1 2) number of entries else 1
7y89	error value for analog outputs	1) 0 … 65535 2) number of entries else 0
Diagnostics and Adminstration Registers		
2008	Status register NANO /JetControl	see seperat description for NANO and JetControl
2011	Module number of a non-intelligent JX2-IO or JX-SIO module with timeout	
2013	Amount of installed non-intelligent JX2-IO and JX-SIO modules	
2015	Index to Module Array	
2016	Module Array	
2070	Amount of installed JX-SIO Modules	
2071	Present system extension size	
7y02	Index to the JX-SIO terminal array	1) 0 … 63 2) 0
7y03	Terminal Array of the JX-SIO Module	1) 0 … 65535 2) number of terminals
7y90	Fault register of the JX-SIO Module	NANO 1) 0 … 65535 2) 0 JetControl 1) 0 … 255 2) 0
7y91	Status register of the JX-SIO Module	NANO 1) 0 … $FFFFFF_{hex}$ 2) 0 JetControl 1) 0 … $FFFFFFFF_{hex}$ 2) 0

Register	Function	1) Value Range 2) Reset Value
7y92	Index to the JX-SIO fault array	1) 0 … 10 2) 0
7y93	Fault array of the JX-SIO Module	1) 0 … FFFFFFFF$_{hex}$ 2) number of entries else 0
7y97	Serial number of the JX-SIO Module	1) 0 … FFFFFFFF$_{hex}$ 2) serial number
7y98	JX-SIO monitoring interval	1) 0 … 255 2) 20
7y99	Version of the JX-SIO software	1) 0 … FFFFFFFF$_{hex}$ 2) version

Glossary

Module number	A unique module number is assigned to each device connected to the system bus. The module numbers of FESTO CPV Direct Valve Terminals range from 70 throug 79.
JX-SIO	The JX-SIO module is the link between the Jetter system bus and the terminals connected to it.
Terminal	A terminal is an expansion module which is connected to the JX-SIO module. The terminal is the link between sensors / actors and the JX-SIO module.
Channel	In most cases, several actors and sensors can be connected to one terminal module. If they are connected to analog input and ouput terminals, the individual actors/sensors are called channels.
Clamp point	The sensor and actuator cables are connected to the terminals by means of plug connectors. Each plug connector accomodates several clamp points for connecting individual cores of cables.
Socket	Up to four plug connectors can be connected to a terminal depending on their width. The plug connectors are snapped onto the terminal sockets.

Tips and Tricks

If no JX-SIO module was found by the controller:

Check baud rate

Check module number; check whether the same module number has been assigned to two JX-SIO modules.

Check wiring