JX6-SB(-I)
Version Update
from V 2.20 to V 2.21

Jetter
Automation. Made easy.

Revision 1.01

November 2008 / Printed in Germany

Revision 1.01

Table of Contents

1 Introduction

Introduction

This chapter shows the history of the JX6-SB(-I) module's operating system versions.

The Purpose of an Operating System Update

What can be done by an operating system update on the JX6-SB(-I) module:

- Expanding the function range
- Fixing software bugs
- Transmitting a certain operating system version, for example at releasing a customer-specific operating system version

Contents

This chapter contains the following topics:

Operating System Update

OS File for Operating System Update

For an operating system update, you will need the following file:

OS File	Description
JX6-SB(-I)_2.21.0.0.os	Operating system file for JX6-SB(-I) of version 2.21.

OS File Download

Jetter AG provide OS files for operating system download on their homepage www.jetter.de. OS files can be found via quicklink on the support site of the JX6-SB(-I) module.

Operating System Update by means of JetSym

Carry out the following steps for an operating system update:

Step	Action
1	OS File Download from www.jetter.de
2	Establishing a connection between PC and controller
3	Executing the menu item Build > Operating System Update in JetSym
4	Selecting the OS file
5	Depending on the controller and on the module, the following items are to be specified: ■ Module number ■ Submodule socket ■ Slave number ■ I/O module number
6	Start the operating system update by ok
7	**Result:** After Power Off / Power On, the new operating system starts.

Overview of the Version Update

V 2.21

The following table gives an overview of the newly added function and the fixed software bugs in operating system version 2.21:

Function	New	Bug
Festo CPX-Terminal:		
Supporting the CPX-CPI Interface	✓	
BWU1821:		
Communication via command interface		✓
Reading and writing analog I/Os		✓
CAN-PRIM:		
User programmable CAN-PRIM interface	✓	
LioN-S:		
Commissioning		✓

2 Expansions

Introduction

The development of Jetter AG constantly expands the functions of the JX6-SB(-I) module. By means of an operating system update, the function range of the module can be expanded. For this, you need ...

- an OS file
- the JetSym software tool
- a connection between PC and controller

Contents

This chapter contains the following topics:

2.1 User-Programmable CAN-PRIM Interface

The CAN-PRIM Interface

The user-programmable CAN-PRIM interface offers the possibility of transmitting and receiving any CAN messages. Processing the CAN messages is done in the application program exclusively.

Applications, e.g.

The following applications can be carried out with the help of the user-programmable CAN-PRIM interface:

- Connection of modules via CAN interface
- Connection of modules via CANopen interface
- ...

Substantial Demands on the Programmer

The functions of the user-programmable CAN-PRIM interface require basic knowledge of the Controller Area Network CAN. Some of them are:

- Structure of a CAN message
- CANopen services

Contents

This chapter contains the following topics:

The Functioning Principle of the CAN-PRIM Interface

Functioning Principle

In a user-programmable CAN-PRIM interface, data exchange between CAN bus and application program takes place via transmit boxes. Each transmit box provides the capacity for a complete CAN message.

The user is supplied with 32 transmit boxes. Each transmit box can be configured either as a sending or as a receiving box. It has got its individual CAN ID.

Technical Data

The technical data of the user-programmable CAN-PRIM interface:

Functioning Principle	Description
CAN ID	11-bit or 29-bit
RTR messages	They are not supported
Number of transmit boxes	32

Activating the CAN-PRIM Interface

The CAN-PRIM interface is activated via bit 2 or bit 3 in MR 2077 *CAN System Bus Special Functions*.

Restrictions on the CAN-PRIM Interface

Restrictions on the Modules to be Connected

For the user-programmable CAN-PRIM interface, the following restrictions apply:

- 9 non-intelligent JX2-I/O modules max. can be connected
- If CANopen modules have been connected, 7 non-intelligent JX2-I/O modules max. can be connected
- If 29-bit CAN identifiers are applied, the serial numbers of the non-intelligent JX2-I/O modules have to start with 2
- at a JC-24x: JX3 modules cannot be connected

CAN Messages to be Transmitted During Boot-Up

During boot-up of the system bus, the connected CAN modules are not permitted to send CAN messages.

Reserved CAN IDs

At synchronous operation of expansion modules at the CAN system bus and at the CAN-PRIM interface, certain CAN IDs have been reserved.

Modules at the System Bus	Reserved CAN IDs
For all modules	0x100, 0x701 - 0x70A, 0x732 - 0x73B, 0x746 - 0x74F
Non-intelligent JX2-I/O modules	0x180 - 0x19F, 0x1A0 - 0x1BF, 0x380 - 0x39F, 0x3A0 - 0x3BF
Intelligent JX2 modules	0x09F - 0x0AF, 0x161 - 0x16F, 0x1D1 - 0x1DF
JX3 modules	0x180 - 0x19F, 0x1A0 - 0x1BF, 0x320 - 0x33E, 0x380 - 0x39F, 0x3A0 - 0x3BF, 0x3E0 - 0x3FE
JX-SIO and third-party modules	0x1C6 - 0x1CF, 0x246 - 0x24F, 0x2C6 - 0x2CF, 0x346 - 0x34F, 0x3C6 - 0x3CF, 0x446 - 0x44F, 0x4C6 - 0x4CF, 0x581 - 0x58A, 0x5B2 - 0x5BB, 0x5C6 - 0x5CF, 0x601 - 0x60A, 0x632 - 0x63B, 0x646 - 0x64F, 0x732 - 0x73B, 0x746 - 0x74F
FESTO CP-FB modules	0x010, 0x110, 0x120, 0x130, 0x140, 0x150, 0x1E0, 0x1F0, 0x250, 0x260, 0x270, 0x350, 0x360, 0x370, 0x3B0
LioN-S modules	0x2E0 - 0x2FE, 0x360 - 0x37E, 0x581 - 0x5A0, 0x601 - 0x620, 0x701 - 0x720
BWU1821	0x281 - 0x29F, 0x301 - 0x31F, 0x481 - 0x49F, 0x501 - 0x51F, 0x5C6 - 0x5CF, 0x646 - 0x647, 0x746 - 0x74F
LJX7-CSL	0x481 - 0x49F, 0x501 - 0x51F, 0x581 - 0x5A0, 0x601 - 0x620, 0x701 - 0x720

Programming the CAN-PRIM Interface

Overview of Registers

For this instruction, the following registers, respectively module registers, are mentioned:

Register	Description
MR 2077	CAN system bus - special functions
MR 10500	Status Register CAN-PRIM
MR 10501	Command Register CAN-PRIM
MR 10502	Box Number
MR 10503	FIFO Occupancy
MR 10510	Box Status
MR 10511	Box Configuration
MR 10512	CAN ID
MR 10513	Number of Data Bytes
MR 10514	Data Byte 0
MR 10515	Data Byte 1
MR 10516	Data Byte 2
MR 10517	Data Byte 3
MR 10518	Data Byte 4
MR 10519	Data Byte 5
MR 10520	Data Byte 6
MR 10521	Data Byte 7

Initialization

Carry out the following steps for the initialization of the CAN-PRIM interface:

Step	Action
1	Set bit 2 = 1 or bit 3 = 1 in MR 2077 *System Bus Special Functions*.
2	Start the system bus.
3	Configure the CAN ID length for all transmit boxes.

If the CAN ID length...	Then ...
is 11 bit	MR 10501 := 8;
is 29 bit	MR 10501 := 9;

Configuring a Transmit Box for Sending

Carry out the following steps for the configuration of a transmit box:

Step	Action
1	Selecting a transmit box MR 10502 := number of the transmit box;
2	Configuring a transmit box as a sending box MR 10511 := 1;
3	Configuring a CAN ID for sending MR 10512 := CAN ID;
4	Activating the box MR 10501 := 1;
5	Result of successful configuring: Bit 0 = 1 in MR 10510

Sending a CAN Message

Carry out the following steps for sending a CAN message:

Step	Action
1	Selecting a transmit box MR 10502 := number of the transmit box;
2	Number of bytes to be sent: MR 10513 := Number of bytes;
3	Writing the data bytes MR 10514 := Data byte 0; MR 10515 := Data byte 1; ... MR 10521 := Data byte 7;
4	Sending the data from the called-up transmit box MR 10501 := 3;
5	Result of successful sending: Bit 3 = 0 in MR 10510

Configuring a Transmit Box for Receiving

Carry out the following steps for configuring a transmit box into a receiving box:

Step	Action
1	Selecting a transmit box MR 10502 := number of the transmit box;
2	Configuring a transmit box as a receiving box MR 10511 := 0;

Step	Action
3	Configuring a CAN ID for receiving MR 10512 := CAN ID;
4	Activating the box MR 10501 := 1;
5	Result of successful configuring: Bit 0 = 1 in MR 10510

Receiving a CAN Message

Carry out the following steps for receiving a CAN message:

Step	Action
1	Checking bit 1 NEWDAT in MR 10500 **If ...** / **Then ...** Bit 1 = 1 in MR 10500 → a CAN message has been received. To be continued with step 2.
2	Selecting the transmit box that has received a new CAN message. MR 10502 := MR 10504;
3	Checking for an overflow ("overrun") of the transmit box. **If ...** / **Then ...** Bit 2 = 1 in MR 10510 → an overflow has occurred.
4	Reading the number of the received bytes Number of bytes := 10513;
5	Reading the received bytes Data byte 0 := MR 10514; Data byte 1 := MR 10515; ... Data byte 7 := MR 10521;
6	Acknowledging the received message MR 10501 := 4;
7	The transmit box is again ready to receive.

Internal Processes of the CAN-PRIM Interface

Introduction

The CAN-PRIM interface processes the following tasks automatically:

- Receiving CAN messages
- Sending CAN messages
- Filtering CAN messages at receiving them

Internal Receiving of CAN Messages

The CAN-PRIM receives a new CAN message as follows:

<table>
<tr><th>Stage</th><th>Description</th></tr>
<tr><td>1</td><td>A valid CAN message has been received by the CAN bus.</td></tr>
<tr><td>2</td><td>The CAN ID agrees with the receiving mask.</td></tr>
<tr><td>3</td><td>The CAN ID agrees with the CAN ID of a transmit box that has been configured for receiving.</td></tr>
<tr><td>4</td><td>
<table>
<tr><th>If in MR 10510 of the transmit box ...</th><th>Then ...</th></tr>
<tr><td>the NEW DAT bit = 0</td><td>becomes NEW DAT bit = 1
To be continued with step 4</td></tr>
<tr><td>the NEW DAT bit = 1</td><td>the OVERRUN bit = 1
The data of the CAN message are discarded</td></tr>
</table>
</td></tr>
<tr><td>5</td><td>The value of MR 10503 FIFO Occupancy is increased by one.</td></tr>
<tr><td>6</td><td>The number of the transmit box is written to MR 10504 FIFO-Data.</td></tr>
<tr><td>7</td><td>In MR 10500, the NEW DAT bit = 1.</td></tr>
</table>

Register Description of the CAN-PRIM Interface

MR 2077

CAN System Bus - Special Functions

Via MR 2077, various special functions of the system bus are released or blocked.

Meaning of the Individual Bits		
Bit 2	**Activating the CAN-PRIM Interface**	
	1 =	Activate the CAN-PRIM interface at the next start-up of the system bus. Expansion modules can be connected to it.
Bit 3	**Activating the CAN-PRIM Interface Only**	
	1 =	Activate the CAN-PRIM interface at the next start-up of the system bus. Expansion modules cannot be connected.

MR 10500

Status Register CAN-PRIM

Via MR 10500, the status of the CAN-PRIM interface can be evaluated.

Meaning of the Individual Bits		
Bit 1	**NEW-DAT**	
	1 =	At least one transmit box that has received a new CAN message.
Bit 2	**ID Length**	
	0 =	CAN IDs with a length of 11 bits are sent/received
	1 =	CAN IDs with a length of 29 bits are sent/received

Module Register Characteristics	
Access	Read
Value following a reset	The CAN-PRIM interface has been activated.

MR 10501

Command Register CAN-PRIM

Via MR 10501, certain commands are transmitted to the CAN-PRIM interface.

Commands of the CAN-PRIM Interface	
1	**Activate the Transmit Box** The transmit box called up from MR 10502 is activated. At activating, a checkup is made, whether the CAN ID of the box has not been reserved by the system bus. Result: Bit 0 = 1 in MR 10510
2	**Deactivate the Transmit Box** The transmit box called up from MR 10502 is deactivated. Result: Bit 0 = 0 in MR 10510
3	**Send a CAN Message** A CAN message is sent with the data of the called-up transmit box.
4	**Clear NEW-DAT Bit** The NEW-DAT bit in MR 10500 is cleared. The called-up transmit box is able to receive CAN messages again. Result: Bit 1 = 0 in MR 10510
5	**Clear the OVERRUN (Overflow) Bit** Clears the OVERRUN bit in MR 10510 of the transmit box. Result: Bit 2 = 0 in MR 10510
6	**Clear Sending Error Bit** Clears the sending error bit in MR 10510 of the called-up transmit box. Result: Bit 3 = 0 in MR 10510
7	**Clear FIFO** Deletes all entries in the FIFO. Result: MR 10503 = 0
8	**Set the Standard ID Length to11 Bits** The ID length for all CAN messages is set to 11 bits. Result: Bit 2 = 0 in MR 10500 MR 10506 := 0 MR 10507 := 0
9	**Set the Standard ID Length to 29 Bits** The ID length for all CAN messages is set to 29 bits. Result: Bit 2 = 1 in MR 10500 MR 10506 := 0 MR 10507 := 0

Module Register Characteristics	
Access	The CAN-PRIM interface has been activated.

MR 10502

Number of the Transmit Box

Via MR 10502, a transmit box is selected. The data of the transmit box can then be accessed via module registers MR 10510 through MR 10521.

Module Register Characteristics		
Values	Number of the Transmit Box	0 ... 31
Access	Reading deletes characters	
Effect	The CAN-PRIM interface has been activated.	

MR 10503

FIFO Occupancy

MR 10503 shows, whether there have been new CAN messages and the number of them.

Module Register Characteristics		
Values	Number of messages received	0 ... 32
Access	Read	
Effect	The CAN-PRIM interface has been activated.	

MR 10504

FIFO Data

MR 10504 shows, in which transmit box a new CAN message has been received. At reading MR 10504, the FIFO is cleared from the value just read. Accordingly, the value of MR 10503 is incremented by one.

Module Register Characteristics		
Values	There are no FIFO data	-1
	Number of the transmit box containing new files	0 ... 31
Access	Reading deletes characters	
Value following a reset	-1	
Effect	The CAN-PRIM interface has been activated.	

MR 10506

Global Receiving Mask

The global receiving mask filters the bits of the received CAN ID. If the bit in the global receiving mask is set, the bit of the received CAN ID is compared with the global receiving ID.

Module Register Characteristics		
Values	for 11-bit CAN IDs	0 ... 0x7FF
	for 29-bit CAN IDs	0 ... 0x1FFFFFFF
Bit = 0	The bit is not compared with MR 10507.	
Bit = 1	The bit is compared with MR 10507.	
Effect	The CAN-PRIM interface has been activated.	

MR 10507

Global Receiving ID

By means of the global receiving ID and MR 10506 *Global Receiving Mask* a range of CAN IDs is set, which is transmitted to the CAN-PRIM interface.

Module Register Characteristics		
Values	for 11-bit CAN IDs	0 ... 0x7FF
	for 29-bit CAN IDs	0 ... 0x1FFFFFFF
Effect	The CAN-PRIM interface has been activated.	

MR 10510

Box Status Register

Via MR 10510, the status of a transmit box can be evaluated.

Meaning of the Individual Bits		
Bit 0	**Valid**	
	1 =	The transmit box is activated
Bit 1	**NEW-DAT**	
	1 =	The transmit box has received a CAN message. Receiving further CAN messages is blocked.
Bit 2	**OVERRUN**	
	1 =	A new CAN message was received, when NEW-DAT was 1.
Bit 3	**Sending error**	
	1 =	At sending a CAN message out of this transmit box, an error has occurred.

Module Register Characteristics	
Access	Read
Effect	The CAN-PRIM interface has been activated.

MR 10511

Box Configuration Register

Via MR 10511, the box can be configured.

Meaning of the Individual Bits		
Bit 0	**Sending / Receiving Box**	
	0 =	Sending box
	1 =	Receiving box

Module Register Characteristics	
Effect	The CAN-PRIM interface has been activated.

MR 10512

CAN ID

If a sending box is applied, a CAN message is sent by this CAN ID.

If a sending box is applied, a CAN message is sent by this CAN ID.

Module Register Characteristics		
Values	for 11-bit CAN IDs	0 ... 0x7FF
	for 29-bit CAN IDs	0 ... 0x1FFFFFFF
Effect	The CAN-PRIM interface has been activated, while the transmit-box has not been activated, which means that bit 0 = 0 in MR 10510.	

MR 10513

Number of Data Bytes

If a sending box is applied, a CAN message is sent by this number of data bytes.

If a receiving box is applied, the number of data bytes having been received in the CAN message is entered.

Module Register Characteristics		
Values	Number of data bytes	0 ... 8
Effect	The CAN-PRIM interface has been activated.	

MR 10514 .. MR 10251

Daba Bytes 0 to 7

If a sending box is applied, a CAN message is sent by these data bytes.

If a receiving box is applied, the data bytes having been received in the CAN message are entered.

Module Register Characteristics		
Values	Data of the data bytes	0 ... 255
Effect	The CAN-PRIM interface has been activated.	

Example of Applying the CAN-PRIM Interface

Task

Via the CAN interface, CAN messages are to be received by the CAN IDs 0x200. After receiving them, a CAN message is to be sent by CAN ID 0x277.

Solution

Via the CAN-PRIM interface, the data are sent and received. For this purpose, a message box for CAN ID 0x200 is installed. Another transmit box is configured as a box of CAN ID 0x277.

Configuration

In this example, the CAN-PRIM interface of a JX6-SB(-I) submodule is made use of. The JX6-SB(-I) submodule is located in socket # 1 on the JC-647.

This means, prefix 310, respectively 31, has to precede the module register numbers of the CAN-PRIM interface.

JetSym ST Program Configuration

The CAN-PRIM interface is configured by the following JetSym ST program.

```
Type
    SYSREG_CANPRIM:
    Struct
        n_State : int;
        n_Command : int;
        n_BoxNumber : int;
        n_FifoNumData : int;
        n_Fifo_Data : int;
        zzDummy10 : int;
        n_GlobalMask : int;
        n_GlobalID : int;
        zzDummy10 : array[2] of int;
        n_BoxState : int;
        n_BoxConfig : int;
        n_BoxCanId : int;
        n_BoxDLC : int;
        to_BoxData : array[8] of int;
    End_Struct;
End_Type;

Var
    n_SysBusSpecial : int at %vl 3102077;
    st_Can_PRIM : SYSREG_CANPRIM at %vl 3110500;
    n_JX6SB_State : int at %vl 111100;
    n_JX6SB_Command : int at %vl 111101;
    to_Data : array[8] of int at %vl 100;
End_Var;
```

```
Task 0
    // activate CAN-PRIM and start CAN system bus
    bit_set(n_SysBusSpecial, 2);
    n_JX6SB_Command := 30;
    When
        bit_clear(n_JX6SB_State, 13)
    Continue;

    // 11-bit CAN ID
    st_CanPrim.n_Command := 8;

    // select box 0
    st_CanPrim.n_BoxNumber := 0;
    // configure the box to ID 0x200
    st_CanPrim.n_BoxCanId := 0x200;
    // configure as receiving box
    st_CanPrim.n_BoxConfig := 0;
    // activate the box
    st_CanPrim.n_Command := 1;
    If
        bit_clear(st_CanPrim.n_BoxState, 0)
    Then
        // CAN ID already used by system bus
    End_If;

    // select box 1
    st_CanPrim.n_BoxNumber := 1;
    // configure the box to ID 0x2FF
    st_CanPrim.n_BoxCanId := 0x2FF;
    // configure as sending box
    st_CanPrim.n_BoxConfig := 1;
    // activate the box
    st_CanPrim.n_Command := 1;
    If
        bit_clear(st_CanPrim.n_BoxState, 0)
    Then
        // CAN ID is already being used by CAN system bus
    End_If;
End_Task;
```

JetSym ST Program - Receiving Data

CAN messages are received by the following JetSym ST program.

```
// waiting for new CAN messages
When
    bit_set(st_CanPrim.n_State, 1)
Continue;

// Read box number out of the Fifo and select the box
st_CanPrim.n_BoxNumber := st_CanPrim.n_FifoData;

// check overrun
If
    bit_set(st_CanPrim.n_BoxState, 2)
Then
    // acknowledge overrun
    st_CanPrim.n_Command := 5;
End_If;

// copy received data
to_Data[0] := st_CanPrim.to_BoxData[0];
to_Data[1] := st_CanPrim.to_BoxData[1];
```

JetSym ST Program - Sending of Data

CAN messages are sent by the following JetSym ST program:

```
// select box 1
st_CanPrim.n_BoxNumber := 1;

// Amount of data bytes = 2
st_CanPrim.n_BoxDLC := 2;
// enter the data to be sent
st_CanPrim.to_BoxData[0] := 12;
st_CanPrim.to_BoxData[1] := 25;

// start sending the CAN message
st_CanPrim.n_Command := 3;

// check sending error
If
    BIT_SET(st_CanPrim.n_BoxState, 3)
Then
    // acknowledge sending error
    st_CanPrim.n_Command := 6;
End_If;
```

2.2 Festo CPX-CP Interface

Introduction

The CPX-CP interface is an expansion module for the CPX terminal of the Festo AG & Co. KG. The CPX-CP interface serves to connect the fitting CPI modules by Festo (CP valve terminals and CPI-I/O modules).

- 192 digital inputs and 192 digital outputs can be directly connected to a CPX-FB14
- Further, up to 4 analog inputs and outputs can be connected

Documentation published by Festo AG & Co. KG

The individual modules of the CPX and CPI system have been specifically described in the documentation published by Festo AG & Co. KG. Giving heed to the safety regulations listed there, as well as only using the modules for their intended purpose are imperative.

Required Hardware and Operating System Versions

The following hardware and operating system versions are required for applying the CPX-CPI interfaces to Jetter AG controllers:

Module	Comment	As of Software Release
JC-24x	Controller	V 3.26
JX6-SB / JX6-SB-I	JX6 submodule	V 2.21
CPX-CP-4-FB	Festo: CPX-CP interface	
CPX-FB14	Festo: CANopen field bus node	

Contents

This chapter contains the following topics:

Restrictions at Using the CAN System Bus

Restrictions at Using the CPX-CP Interface

The usage of the CPX-CP interface at the CPX-FB14 is subject to the following restrictions:

- The required hardware and operating system versions have to be kept to.
- The CPX-FB14 supports one CPX-CP interface as a maximum.
- In the system bus, the CPX-FB14 occupies up to three I/O module numbers. These I/O module numbers are not available in other modules.
- The number of the analog I/Os at the CPX-FB14 is reduced.

Restrictions at Connecting Analog I/Os

At applying the CPX-CPI interface, the number od the analog inputs and outputs to be connected is reduced.

If at the CP string 1 or 2 ...	Then the number of ...
CP / CPI output modules are connected,	the analog outputs is reduced to 8
CP valve terminals are connected,	the analog outputs is reduced to 8
CP / CPI input modules are connected,	the analog inputs is reduced to 8

If at the CP string 3 or 4 ...	Then the number of ...
CP / CPI output modules are connected,	the analog outputs is reduced to 4
CP valve terminals are connected,	the analog outputs is reduced to 4
CP / CPI input modules are connected,	the analog inputs is reduced to 4

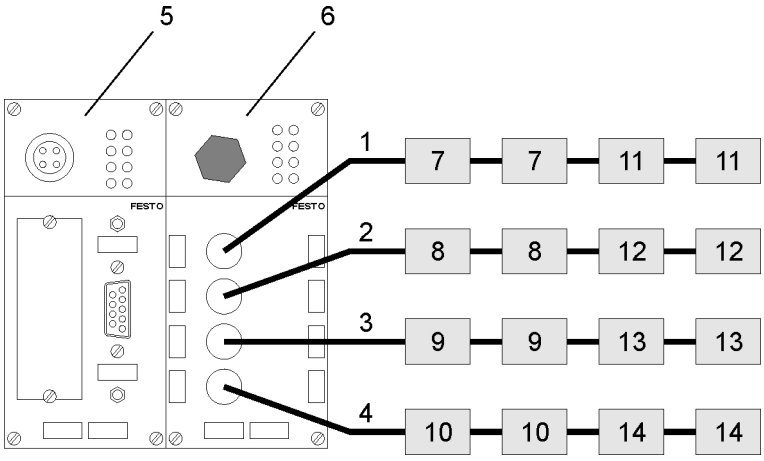
Restrictions at Connecting CANopen Modules

By using the additional I/O module numbers, the maximum number of CANopen modules that can be connected to the CAN system bus is reduced. Each additional I/O module number causes the number of CANopen modules to be reduced by one.

Access to the Modules at the CPX-CP Interface

I/O Assignment of the Modules at the CPX-CP Interface

The digital inputs and outputs of the modules connected to the CPX-CPI interface are mapped by the I/O numbers of the following I/O modules:



Number	Element	I/O Numbers of JC-24x
1 ... 4	CPI string 1 ... 4	-
5	CPX FB14	e.g. I/O module number 70
6	CPX-CP-4-FB	-
7	CP / CPI output modules	OUT 7101 ... 7132
8	CP / CPI output modules	OUT 7133 ... 7164
9	CP / CPI output modules	OUT 7201 ... 7232
10	CP / CPI output modules	OUT 7233 ... 7264
11	CP / CPI input modules	IN 7101 ... 7132
12	CP / CPI input modules	IN 7133 ... 7164
13	CP / CPI input modules	IN 7201 ... 7232
14	CP / CPI input modules	IN 7233 ... 7264

I/O Numbers of JC-647

The I/Os at the JC-647 connected with the JX6-SB(-I) submodule are numbered in analogy with the I/O numbering at the JC-24x. The input and output numbers have just got to be prefixed. The prefix corresponds to the submodule slot number + 1.

Assigning the I/O Module Numbers

A CPX terminal with a CPX-CPI interface occupies up to three I/O module numbers in the CAN system bus. The digital I/Os are represented by the I/O module numbers according to the following rules:

- the three I/O module numbers max. are consecutive
- the local I/Os of the CPX terminal are represented by the first I/O module number
- the I/Os at the CPI strings 1 and 2 are represented by the subsequent I/O module number

- the I/Os at the CPI strings 3 and 4 are represented by the next but one I/O module number

Entries in the Module Array

The numbers of the additionally occuppied I/O modules are entered into the module array as a virtual CPX-FB14.

Element	Module Code
CPX-FB14	66
CPX-FB14 virtual	73

3 Fixed Software Bugs

Introduction

This chapter describes the software bugs which have been fixed in the new operating system release.

Contents

This chapter contains the following topics:

Reading and Writing Analog I/Os at the BWU1821

BWU1821

The BWU1821 is a gateway between AS interface and CANopen of the Bihl+Wiedemann GmbH. It can be directly connected to the Jetter CAN system bus.

Appearance of the Error

Reading and writing of AS interface slaves by means of analog inputs and outputs renders wrong results.

Releases Concerned

The error occurs in the following releases:

Operating System Release	JX6-SB(-I) JC-24x JM-D203-JC24x	< 2.21.0.00 < 3.26.0.00 < 1.13.0.00
Hardware Revision	not relevant	
Configuration or Operating Mode	not relevant	
Comment	600	

Remedy / Workaround

Carry out the following steps for a workaround:

Step	Action
1	Configuring a NODE ID of the BWU1821 on 70

Bugfix in the Following Updates

The error has been fixed as of the following releases:

Operating System Release	JX6-SB(-I) JC-24x JM-D203-JC24x	2.21.0.00 3.26.0.00 1.13.0.00
Hardware Revision	not relevant	
Configuration or Operating Mode	not relevant	

Writing Commands to the BWU1821

BWU1821

The BWU1821 is a gateway between AS interface and CANopen of the Bihl+Wiedemann GmbH. It can be directly connected to the Jetter CAN system bus.

Appearance of the Error

Commands of a length greater than 4 bytes cannot be transmitted to the BWU1821. The following functions, for example, are affected by this error:

- Reading parameters of AS interface slaves according to profile 7.4
- Writing parameters to AS interface slaves according to profile 7.4
- Reading input data of AS interface slaves according to profile 7.4
- Writing output data to AS interface slaves according to profile 7.4
- ...

Releases Concerned

The error has been fixed as of the following releases:

Operating System Release	JX6-SB(-I) JC-24x JM-D203-JC24x	< 2.21.0.00 < 3.26.0.00 < 1.13.0.00
Hardware Revision	not relevant	
Configuration or Operating Mode	not relevant	
Comment	610	

Remedy / Workaround

There is no remedy to be applied to the releases concerned.

Bugfix in the Following Updates

The error has been fixed as of the following releases:

Operating System Release	JX6-SB(-I) JC-24x JM-D203-JC24x	2.21.0.00 3.26.0.00 1.13.0.00
Hardware Revision	not relevant	
Configuration or Operating Mode	not relevant	

Initializing by means of LioN-S Modules

Appearance of the Error

In case of CAN system bus configurations consisting of LioN-S and JX2-/JX3 modules, not all connected JX2-/JX3 modules are commissioned.

Releases Concerned

The error occurs in the following releases:

Operating System Release	JX6-SB(-I) JC-24x JM-D203-JC24x	< 2.21.0.00 < 3.26.0.00 < 1.13.0.00
Hardware Revision	not relevant	
Configuration or Operating Mode	not relevant	
Comment	698	

Remedy / Workaround

Carry out the following steps for a workaround:

Step	Action
1	Start the CAN system bus
2	Re-start the CAN system bus without de-energizing the expansion modules

Bugfix in the Following Updates

The error has been fixed as of the following releases:

Operating System Release	JX6-SB(-I) JC-24x JM-D203-JC24x	2.21.0.00 3.26.0.00 1.13.0.00
Hardware Revision	not relevant	
Configuration or Operating Mode	not relevant	