JC-350

Version Update
from V 1.05 to V 1.08

Version Update

Jetter

Revision 1.01

Table of Contents

1 Introduction

Introduction

This chapter shows the history of OS versions for the controller JC-350.

Operating System Update - Why?

An OS update allows you to:

- add new functions to your controller
- fix software bugs
- make sure your controller is working with a definite OS version, for example, if a definite OS version has been released for a certain customer

Contents

Operating System Update

OS File for Updating the Operating System

For updating the OS the following file is needed:

OS File	Description
JC-350_1.08.0.0.os	OS file for JC-350 with version 1.08

Downloading the OS File

Jetter AG make operating system files available for download from their **homepage at http://www.jetter.de**. OS files can be found in the support area or on the page of the JC-350 controller via quicklink.

Operating System Update by means of JetSym

To update your OS proceed as follows:

Step	Action
1	Download the OS file from www.jetter.de
2	Establish a connection between PC and controller
3	In JetSym: Select menu item "Build -> Update OS" or Click on the button "OS Update" in the CPU window of the hardware manager
4	Select the OS File
5	Initiate the OS update by clicking OK
6	**Result:** Following Power OFF / Power ON the new OS is launched.

Minimum Requirements

For programming a JC-350 with version 1.08 JetSym 4.1 or higher is required.

JC-350 Version Update - Overview

V 1.04

The following table gives an overview of newly added features and fixed software bugs in OS version 1.04:

Function	New	Fixed
JX2 System Bus:		
Register overlaying for digital inputs/outputs	✓	
Support of JX-SIO modules and third-party CANopen devices	✓	
JX3 system bus:		
Register overlaying for digital inputs/outputs	✓	
System bus special registers for status and control	✓	
Operating System Update:		
Via FTP: On completion notification the OS has actually been stored.		✓
Updating a JX2 slave module while registers are being accessed blocks communication		✓
Application program:		
Task switch could fail to happen		✓
Error signal in case of invalid file "/app/start.ini"		✓
Display commands:		
Redirection to JX2-SER1 works only if JX2-PRN1 has been configured		✓

V 1.05

The following table gives an overview of newly added features and fixed software bugs in OS version 1.05:

Function	New	Fixed
JX2 System Bus: V1.05.0.00		
AS interface gateway BWU1821 is supported	✓	
Frequency inverter 8200 vector is supported	✓	
JetMove 1xx is not detected during boot process		✓
Automatic baud rate recognition does not work reliably for some of the baud rates and configurations of IP67 modules.		✓
Repetition counter does not work when polling I/O modules		✓
AutoCopy function:		
Automatic copying of controller data	✓	
Application program:		
Pending cyclic tasks are started immediately after Taskunlock	✓	
For function pow(x,y) a floating point number can be entered as exponent	✓	

Function	New	Fixed
Cyclic tasks can be debugged	✓	
Length of project and program names > 39 characters		✓
Restart of an elapsed timer		✓
The value returned by DateTimeDecode() was always 1 day short of the actual day.		✓
DateTimeEncode and -IsValid might return the value TRUE irrespective of an invalid date		✓
User registers:		
The register type can be set up without having to start the application program	✓	
Displays and HMIs:		
A floating point value can be used as default for UserInput	✓	
The default value for UserInput is not displayed correctly		✓
It is not possible to enter LED register numbers		✓

V 1.08

The following table gives an overview of newly added features and fixed software bugs in OS version 1.08:

Function	New	Fixed
System configuration:		
System rights for configuration file	✓	
JX2 System Bus: V1.11.0.00		
Timeout after CAN-PRIM message		✓
Registers of LJX7-CSL modules		✓
Write access to analog outputs of CANopen® modules		✓
State of digital inputs when the controller is powered on		✓
Digital outputs on JX-SIO or CANopen® modules		✓
Input/output 64 on JX-SIO or CANopen® modules		✓
User-programmable CAN Interface		✓
Application program:		
NetCopyList Functions	✓	
StrCopy()		✓
Crash in the case of "invalid" application program		✓
NetCopyVarFromReg()		✓
JX3 system bus:		
Module registers for digital I/Os	✓	
Displays and HMIs:		
UserInput()		✓

2 New Features

Introduction

This chapter describes the features which have been added or enhanced in the new software release.

Contents

2.1 Various New Features and Modifications

Introduction

This chapter covers the new features and modifications

Contents

Access rights for access to configuration memory

Introduction	The configuration file "/System/config.ini" is used to access the configuration memory of the controller.
Obsolete Function	In the case of an FTP connection to the controller the user must have administrator rights to be able to access the configuration file.
New Function	In the case of an FTP connection to the controller the user must have administrator or system rights to be able to access the configuration file.
Reason for this change	This way, a user without administrator rights is allowed to access the configuration memory on the controller via file system.

Registers for digital JX3-I/O modules

Obsolete Function	Inputs and outputs of digital JX3 modules (JX3-DI16, JX3-DIO16, JX3-DO16) can be accessed via individual inputs/outputs (10000mm01 through 10000mm16) or via registers of combined inputs/outputs (100004xxx).
New Function	In addition to the existing functions, the 16 inputs can be read out via module register 2. Outputs can be read out and values can be written to them via module register 3. Register number of inputs: 100mm0002 Register number of outputs: 100mm0003
Reason for this change	Standardization of registers and types of access.

System function NetCopyList using STX variables

Introduction

No changes have been made to existing functions. Only system function 150 (configuring NetCopyList) has been enhanced (red). System functions 151 and 152 remain unchanged.

Obsolete Function

Only registers (%VL) can be used for local data.

New Function

Registers (%VL) or STX variables can be used for local data.

Restrictions

As with all system functions, the function parameters and the result of the function have still to be stored to registers.

If STX variables are to be used for function parameters and the result of the function, STX functions of the function group NetCopyList (NetCopyList, NetCopyListInit, NetCopyListConfig, NetCopyListSend, and NetCopyListDelete) must be used.

Function Declaration

```
Systemfunction(150, &StructNetCopyList, &RegResult);
```

Parameter	Function
StructNetCopyList	Structure consisting of the type NCL_HEADER and, per communication unit, of the type NCL_ELEMENT.
RegResult	Number of the register to which the result of this function will be stored.

Type declaration NCL_HEADER

```
Type
  NCL_HEADER:
  Struct
     IPaddress      : Int;
     IPport         : Int;
  End_Struct;
End_Type;
```

Type declaration NCL_ELEMENT

```
Type
  NCL_ELEMENT:
  Struct
     Command        : Int;
     Mode           : Int;
     NoOfRegs       : Int;
     LocalAddr      : Int;
     RemoteAddr     : Int;
  End_Struct;
End_Type;
```

Function Parameters

Parameter	Value	Comment
Header		
IPaddress		IP address of the remote PLC
IPport		Port number of the remote PLC
Communication unit 1		
Command	1	Read access: Copying remote registers into local registers
	2	Write access: Copying local registers into remote registers
	3	Read Access: Copying remote registers into local STX variables
	4	Write access: Copying local STX variables into remote registers
Mode	1	Autoincrementation of remote register number
	2	Autodecrementation of remote register number
NoOfRegs	1..64	Quantity of registers
LocalAddr		Command 1 and 2: Local register number
		Command 3 and 4: Local address of STX variable
RemoteAddr		Number of remote register
...	...	...

Result of the Function

This function will produce one of the following results:

Result of the Function	
>0	Handle associated with the list
-1	all lists are already being used; no available list has been found
-2	all communication units are already being used; no available communication units have been found
-3	Empty list
-4	Invalid list
-5	Invalid IP address
-6	Invalid instruction
-8	Invalid mode
-9	Number of registers too large / invalid address of local variables
-10	Maximum size of requested transmit buffer exceeded
-11	Maximum size of requested receiving buffer exceeded
-20	no JetIP V1.1 available

2.2 NetCopyList

Introduction

The NetCopyList functions can be used to combine several read/write accesses to registers located on another controller in one Ethernet telegram. In the following cases this results in a significant performance increase:

- If several registers are to be transmitted which are distributed across the memory;
- If registers are to be read and written in one go.

Minimum Requirements

The NetCopyList can be used starting from JetSym version 4.1.3.

Operating Principle

- Communication units are for defining which register areas are to be transmitted and whether they are to be read or written.
- One or several communication units can be combined in a list.
- Several lists of different length can be configured.
- When sending a list, all communication units of this list are sent in an Ethernet telegram.

Technical Data

Property	Value
Max. number of lists	10
Total number of communication units	500
Maximum number of registers in a list	64

Contents

Programming the NetCopyList Functions

Introduction

The NetCopyList functions which are used for programming this feature are included in the programming language of the controller. To program this feature proceed as follows:

Step	Action
1	Initializing the NetCopyList feature (once on program start)
2	Configuring the list(s)
3	Transmitting data by sending the list(s)
4	Optionally deleting the list(s)

Overview of Functions

The following functions are available:

Number	Function	Description
1	NetCopyListInit	Initialization must be carried out at least once each time the application program is launched.
2	NetCopyListConfig	Configuration of a list
3	NetCopyListSend	Sending of a list
4	NetCopyListDelete	Deleting of a list
5	NetCopyList	Processing of functions 2 through 4 in one call

Predefined Data Types

The following data type has been predefined for a communication unit. It can be used in the application program to create the parameter list for functions 2 and 5:

```
Type
    TNetCopyListElement : Struct
        nCommand    : Int;
        nMode       : Int;
        nCount      : Int;
        nLocalReg   : Int;
        nRemoteReg  : Int;
    End_Struct;

    TNetCopyListL   : Array Of TNetCopyListElement;
End_Type;
```

Communication Unit

Each communication unit is for defining one network instruction Several communication units add up to a list of network instruction. Die communication units of a list must be stored to the memory one after another. The parameters have the following meaning:

TNetCopyListElement	
nCommand	
0 =	End of list
1 =	Copy values from remote registers into local registers
2 =	Copy values from local registers into remote registers
3 =	Copy values from remote registers into local variables
4 =	Copy values from local variables into remote registers
nMode	
0 =	Target address is not changed
1 =	Autoincrement the target address
2 =	Autodecrement the target address
nCount	
1 ... 64	Number of registers to be copied
nLocalReg	
nCommand = 1 or 2	Number of the first local register
nCommand = 3 or 4	Address of the first local variable
nRemoteReg	
	Number of the first remote register

Configuring a List

Introduction

Before a list can be used, it must be configured.

Function Declaration

```
Function NetCopyListConfig(IPAddr : int,
                           IPPort : int,
                  const ref List : TNetCopyListL) : int;
```

Function Parameters

Description of function parameters:

Parameter	Value	Comment
IPAddr	Valid IP address	Address of the device to which the list is to be sent
IPPort	Valid IP port	
List	Variable address	Start address of the communication unit list

Result of the Function

This function will produce one of the following results:

Result of the Function	
> 0	Handle associated with this list; this value has to be stored since it will be needed as parameter for sending and deleting the list.
-1	All lists are already being used; no available list has been found
-2	All communication units are already being used; no available communication units have been found
-3	Empty list
-4	Invalid list
-5	Invalid IP address
-6	Invalid instruction
-8	Invalid mode
-9	Maximum number of registers exceeded
-10	Maximum size of requested transmit buffer exceeded
-11	Maximum size of requested receiving buffer exceeded

Using the Function

This function can be used and its result be assigned to a variable for further utilization in the following way:

```
hList := NetCopyListConfig(IP#192.168.25.123, 50000,
                           NetCopyListParam);
```

Operating Principle

The controller processes this function in the following stages:

Stage	Description
1	The controller creates a new list in its internal memory.
2	The controller reads the communication units from the transmitted variable and adds instructions into the list until a communication unit is detected where `nCommand = 0` or an error has occurred.
3	<table><tr><th>If ...</th><th>... Then ...</th></tr><tr><td>no error has occurred</td><td>the function is terminated and a positive value is returned as handle for further access to the list</td></tr><tr><td>an error has occurred</td><td>the function is terminated and a negative value is returned</td></tr></table>

Sending a List

Introduction

This function is for sending a previously configured list.

Restrictions

Tasks which are sending lists must not be started by the application program using `TaskBreak` or be restarted using `TaskRestart` while the controller is processing one of these functions. Failure to take this into account may result in data loss upon receiving.

Function Declaration

```
Function NetCopyListSend(Handle : int) : int;
```

Function Parameters

Description of function parameters:

Parameter	Value	Comment
Handle	Handle associated with the list	Return value when configuring the list

Result of the Function

This function will produce one of the following results:

Result of the Function	
0	No error
-7	Invalid handle
-12	Error during data transfer

Using the Function

This function can be used and its result be assigned to a variable for further utilization in the following way:

```
nResult := NetCopyListSend(hList);
```

Operating Principle

The controller processes this function in the following stages:

Stage	Description
1	The controller compiles an Ethernet telegram with the values to be written and sends it to the communication partner.
2	The task waits at this instruction until it receives a response and assigns the received values to the local variables/registers.
3	The function stores the result to the specified variables.

Deleting a List

Introduction

This function is for deleting a previously configured list.

Function Declaration

```
Function NetCopyListDelete(Handle : int) : int;
```

Function Parameters

Description of function parameters:

Parameter	Value	Comment
Handle	Handle associated with the list	Return value when configuring the list

Result of the Function

This function will produce one of the following results:

Result of the Function	
0	No error
-7	Invalid handle

Using the Function

This function can be used and its result be assigned to a variable for further utilization in the following way:

```
nResult := NetCopyListDelete(hList);
```

Operating Principle

The controller processes this function in the following stages:

Stage	Description
1	The controller deletes the list from its internal memory.

Configuring, Sending and Deleting a List

Introduction

This function consecutively invokes the above mentioned functions for configuring, sending and deleting a list.

Function Declaration

```
Function NetCopyList(IPAddr : int,
                     IPPort : int,
              const ref List : TNetCopyListL) : int;
```

Function Parameters

Description of function parameters:

Parameter	Value	Comment
IPAddr	Valid IP address	Address of the device to which the list is to be sent
IPPort	Valid IP port	
List	Variable address	Start address of the communication unit list

Result of the Function

This function will produce one of the following results:

Result of the Function	
> 0	Handle associated with this list; this value has to be stored since it will be needed as parameter for sending and deleting the list.
-1	All lists are already being used; no available list has been found
-2	All communication units are already being used; no available communication units have been found
-3	Empty list
-4	Invalid list
-5	Invalid IP address
-6	Invalid instruction
-8	Invalid mode
-9	Maximum number of registers exceeded
-10	Maximum size of requested transmit buffer exceeded
-11	Maximum size of requested receiving buffer exceeded
-12	Error during data transfer

Using the Function

This function can be used and its result be assigned to a variable for further utilization in the following way:

```
nResult := NetCopyList(IP#192.168.25.123, 50000,
                       NetCopyListParam);
```

Operating Principle

The controller processes this function in the following stages:

Stage	Description
1	The controller invokes the function `NetCopyListConfig`.
2	The controller invokes the function `NetCopyListSend`.
3	The controller invokes the function `NetCopyListDelete`.

Related Topics:

- **Configuring a List** on page 18
- **Sending a List** on page 20
- **Deleting a List** on page 21

Sample Program: NetCopyList

Task

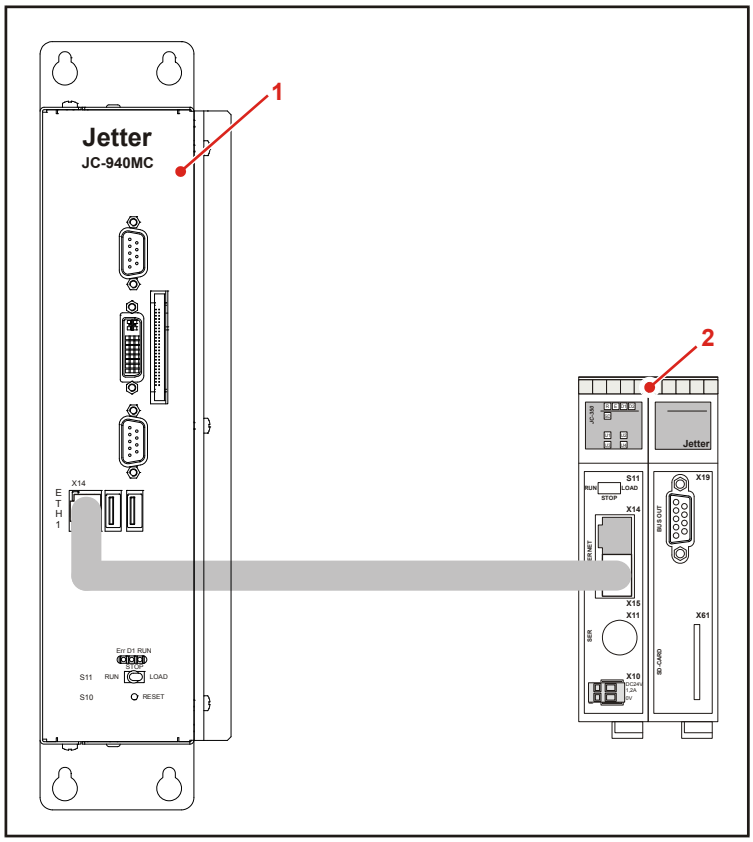
The register and variable contents located on a JetControl 940MC are to be copied into the registers of a JetControl 350. Furthermore, register contents of the JC-350 are to be read out by JC-940MC and copied into local registers and variables. This task is a cyclic one.

Solution

In the application program of JC-940MC a NetCopyList is configured, which reads out and writes to JC-350 registers in one telegram.

Sample Configuration

This example is based on the following configuration:



Number	Description	Function
1	JC-940MC	Controller
2	JC-350	Controller

Due to the platform-independent implementation of the NetCopyList function this sample program can be used for other configurations simply by changing the register numbers.

JetSym STX Program

```
#Include "Platforms.stxp"
```

```
Const
    cNetCopyReadReg     = 1;
    cNetCopyWriteReg    = 2;
    cNetCopyReadVar     = 3;
    cNetCopyWriteVar    = 4;
End_Const;

Var
    nLocalVar        : Int;
    anLocalVar       : Array[5] Of Int;
    LocalReg         : Int At %VL 100;
    aLocalReg        : Array[3] Of Int At %VL 200;

    NetCopyListParam : Array[5] Of TNetCopyListElement;
    hList, nRes      : Int;
End_Var;

Task tNetCopyListText Autorun

    NetCopyListInit();

    // setup one list with four elements
    NetCopyListParam[0].nCommand   := cNetCopyReadReg;
    NetCopyListParam[0].nMode      := 1;
    NetCopyListParam[0].nCount     := 1;
    NetCopyListParam[0].nLocalReg  := &LocalReg;
    NetCopyListParam[0].nRemoteReg := 1000100;

    NetCopyListParam[1].nCommand   := cNetCopyWriteReg;
    NetCopyListParam[1].nMode      := 1;
    NetCopyListParam[1].nCount     := 3;
    NetCopyListParam[1].nLocalReg  := &aLocalReg;
    NetCopyListParam[1].nRemoteReg := 1000200;

    NetCopyListParam[2].nCommand   := cNetCopyReadVar;
    NetCopyListParam[2].nMode      := 1;
    NetCopyListParam[2].nCount     := 5;
    NetCopyListParam[2].nLocalReg  := &anLocalVar;
    NetCopyListParam[2].nRemoteReg := 1000300;

    NetCopyListParam[3].nCommand   := cNetCopyWriteVar;
    NetCopyListParam[3].nMode      := 1;
    NetCopyListParam[3].nCount     := 1;
    NetCopyListParam[3].nLocalReg  := &nLocalVar;
    NetCopyListParam[3].nRemoteReg := 1000400;

    // terminate the parameter list
    NetCopyListParam[4].nCommand   := 0;
```

```
    // configure the list
    hList := NetCopyListConfig(IP#192.168.10.208,
                               50000,
                               NetCopyListParam);

    Loop
        If hList > 0 Then
            // send the list
            nRes := NetCopyListSend(hList);
        End_If;
        Delay(T#1s);
    End_Loop;
End_Task;
```

3 Fixed Software Bugs

Introduction

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

Contents

UserInput() works only in the case of "Autorun"

Effects of this Bug

If a UserInput() instruction is used in a task which has not been started by the task attribute "Autorun" during program launch, the controller JC-350 immediately aborts the UserInput() by throwing the exception USER_INPUT_BREAK.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	968	

Remedy / Workaround

Use the instruction UserInput() only in tasks which are started by the task attribute "Autorun" during program launch.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

The result of UserInput() doesn't match the entered value.

Effects of this Bug

If a number entered via UserInput(() on an HMI exceeds 6 digits, the UserInput() return value sometimes does not correspond to the entered value.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Remedy / Workaround

There is not remedy/workaround for affected versions/revisions.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

The controller crashes in the case of a StrCopy() instruction

Effects of this Bug

If a StrCopy() instruction exceeds 255 characters or has a negative length, the controller JC-350 crashes.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340	< 1.08.0.00
	JC-350	< 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	1123	

Remedy / Workaround

Make sure that the length of the StrCopy() function is between 0 and 255 characters.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340	1.08.0.00
	JC-350	1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Input/output 64 on JX-SIO or CANopen® modules does not work

Effects of this Bug

Access to input/output 64 on a JX-SIO or CANopen® module connected to the JX2 system bus does not work. All other inputs/outputs work properly.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	990	

Remedy / Workaround

Use the corresponding bits in such registers which are overlaid over inputs or outputs.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

The second operating system update of the controller does not work

Effects of this Bug

If the OS of the controller is updated twice in direct succession, the second update is aborted during upload.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340	< 1.08.0.00
	JC-350	< 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Remedy / Workaround

Re-boot the controller following each OS update. This can be done by power cycling the controller.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340	1.08.0.00
	JC-350	1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

The controller crashes after a program download has been aborted

Effects of this Bug

If an application program download has been aborted, for example because of data transmission problems, the controller JC-350 sometimes crashes when attempting to launch the invalid application program.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	1.02.0.00 - 1.05.0.00 1.02.0.00 - 1.05.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Remedy / Workaround

There is no remedy/workaround for affected versions/revisions. Program download errors hardly ever occur.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

When using the NetCopyVarFromReg() instruction, other variables get overwritten, too.

Effects of this Bug

If in the case of the instruction NetCopyVarFromReg() the number of bytes to be transmitted does not correspond to a multiple of 4, it may happen that variables get overwritten which are not the destination of the instruction NetCopyVarFromReg().

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	909	

Remedy / Workaround

Specify the number of bytes to be transmitted as a multiple of 4 (4, 8, 12, etc.).

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

If in the case of the instruction NetCopyVarFromReg() the number of bytes to be transmitted does not correspond to a multiple of 4, network access is not carried out and the instruction NetCopyVarFromReg() is aborted issuing error message "-2".

Write access to an input results in a stack overflow

Effects of this Bug

If the application program contains numerous write access attempts to inputs, the exception STACK_OVERFLOW occurs in this task. If this exception is not intercepted in the application program, the task stops.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Remedy / Workaround

Do not program write accesses to inputs in the application program.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Write access to an input from the application program is still impossible, merely the stack overflow does no longer occur.

The controller crashes after application program download

Effects of this Bug

If in an application program a lot of individual registers have been defined in the %VL area and if these registers are not consecutively stored to the memory, it may happen that the controller JC-350 crashes after this program is uploaded.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340	1.02.0.00 - 1.05.0.00
	JC-350	1.02.0.00 - 1.05.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	1076	

Remedy / Workaround

Locate remanent variables in the %RL area or combine registers within the %VL area in structures or arrays.

Bug Fix

It is not possible to completely fix this bug by an OS update, only the likelihood of its occurrence can be reduced. Starting from JetSym revision 4.2 this bug has been fixed:

OS version	JC-340	1.08.0.00
	JC-350	1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

User-programmable CAN Interface does not work

Effects of this Bug

The user-programmable CAN interface (CAN-PRIM) cannot be used.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Remedy / Workaround

There is no remedy/workaround for affected versions/revisions.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Updating digital outputs on JX-SIO or CANopen® modules

Effects of this Bug

If digital outputs of a JX-SIO or CANopen® module are updated in quick succession, it may happen that the module switches to pre-operational state. In this state, outputs and inputs can no longer be updated. The controller JC-350 reports a timeout.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340	< 1.08.0.00
	JC-350	< 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	1084	

Remedy / Workaround

The workaround for this bug is as follows:

Step	Action
1	Insert into the JetSym STX program a *Delay(T#2ms);* between two write accesses to digital outputs.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340	1.08.0.00
	JC-350	1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

All digital inputs on the JX2 system bus are in state 0 when the controller is powered on

Effects of this Bug

When the controller is powered up, all digital inputs on the JX2 system bus are in state 0 irrespective of the actual input state on the module.

The following modules connected to the JX2 system bus are affected by this bug:

- JX2-I/O modules, digital inputs 200000201 ... 200002416
- CANopen® modules, digital inputs 200007001 ... 200007964
- IP67-I/O modules, digital inputs 200000201 ... 200002416

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	1099	

Remedy / Workaround

The workaround for this bug is as follows:

Step	Action
1	Change the state of a digital input on the module (edge change). Result: The input state of the module is updated by the controller.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Write access to analog outputs of CANopen® modules

Effects of this Bug

No write access to analog outputs of CANopen® modules in registers 200006x68 ... 200006x71 possible.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	1160	

Remedy / Workaround

There is no remedy for affected versions/revisions.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

Timeout when sending of messages via CAN-PRIM

Effects of this Bug

When sending a CAN message via CAN-PRIM interface, the controller JC-350 issues a timeout error message. This error message signals the user that a timeout has occurred when trying to access one of the following modules:

- CANopen® modules
- IP67-I/O modules

Read access to digital inputs and write access to digital outputs of the affected module is no longer possible.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	1103	

Remedy / Workaround

There is no remedy for affected versions/revisions.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	

No access to registers of LJX7-CSL modules

Effects of this Bug

No read or write access to registers R 200003xxz und R 200007xzz of LJX7-CSL modules is possible. These registers are used to configure additional functions of LJX7-CSL modules.

Affected Versions/Revisions

The following versions/revisions are affected by this bug:

OS version	JC-340 JC-350	< 1.08.0.00 < 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	
Note	1161	

Remedy / Workaround

There is no remedy for affected versions/revisions.

Bug Fix

Starting from the following versions/revisions this bug has been fixed:

OS version	JC-340 JC-350	1.08.0.00 1.08.0.00
Hardware revision	not applicable	
Configuration or operating mode	not applicable	