Fuji Flex Driver Help

Table of Contents

Fuji Flex Driver Help

Help version 1.012

CONTENTS

Overview

The Fuji Flex Driver provides an easy and reliable way to connect Fuji Flex devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with Fuji Flex N series devices.

Device Setup

Supported Devices

Fuji Flex N Series PLCs-NB0, NB1, NB2, NB3, NJ, and NS

Communication Protocol

Fuji Computer Link

Supported Communication Parameters

Baud Rate-300, 600, 1200, 2400, 4800, 9600, and **19200**
Parity - None, Even, and **Odd**
Data Bits-7 and **8**
Stop Bits-**1** and 2

Note: The default values are shown in **bold**.

Ethernet Encapsulation

This driver supports Ethernet Encapsulation. Ethernet Encapsulation allows the driver to communicate with serial devices attached to an Ethernet network using a terminal server. Ethernet Encapsulation mode is invoked by selecting it from the COM ID dialog on the channel properties page. More help on Ethernet Encapsulation can be found in the main OPC Server help file.

Device IDs

Valid Device IDs range from 0 to 31.

Flow Control

When using an RS232/RS422 converter, the type of flow control that is required will depend upon the needs of the converter. Some converters do not require any flow control and others will require RTS flow. Consult the documentation of the converter to determine what its flow requirements are.

Cable Diagrams

Communication can be directly through the loader port or a communication unit. **See Also:** **Loader Port Cable Diagram** and **RS-1 Communication Unit Cable Diagram**.

Modem Setup

This driver supports modem functionality. For more information, please refer to the topic "Modem Support" in the OPC Server Help documentation.

Interface Options

Communication with an N series PLC may be directly through the loader port, or through an optional communications unit such as the NB-RS1. If devices on a given channel are equipped with communications units, check the "Using communication unit" box on the channel’s "Interface Options" wizard page or property page.

Networking multiple N series PLCs requires the use of communication units and an RS-485 line. There should be only one device on a channel if communication is through the loader port or through a communications unit with an RS-232C line.

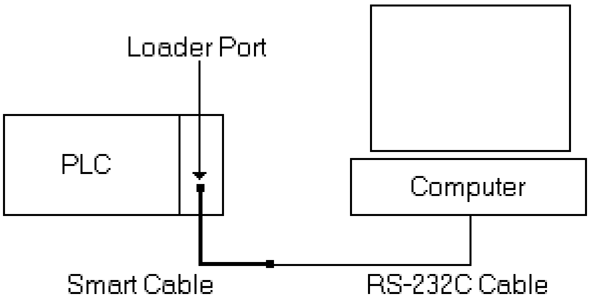
If communication units are used, they must be configured to use ":" (0x3A) for the start code, a carriage return (0x0D) for the end code, and no BCC.

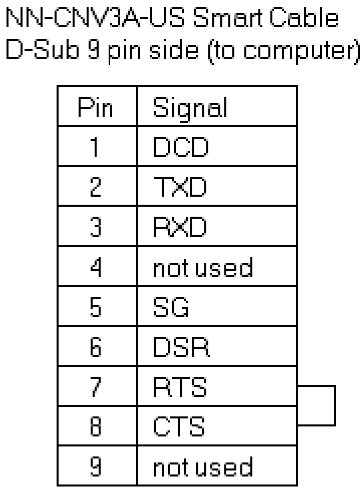
Note: For more information, refer to **Cable Diagram**.

Block Size

The block size is the number of bytes that may be requested from a device at one time. This setting allows you to refine the performance of the driver. If a large number of consecutive data points are being read, a large block size may improve performance, whereas if a few scattered data points are being read, a smaller block size may improve performance. Block sizes range: 16, 32, 64, or 128. 64 is the default setting.

Loader Port Cable Diagram

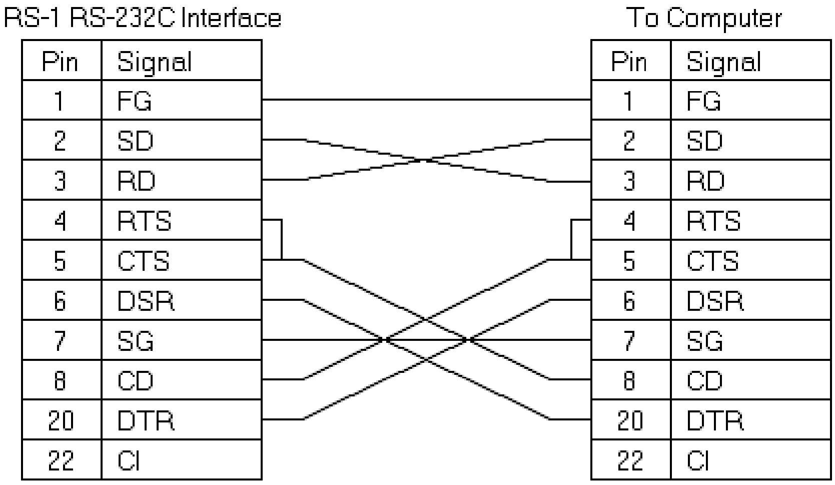



NN-CNV3A-US Smart Cable
D-Sub 9 pin side (to computer)

Pin	Signal
1	DCD
2	TXD
3	RXD
4	not used
5	SG
6	DSR
7	RTS
8	CTS
9	not used

RS-1 Communication Unit Cable Diagram

1:1 Network using RS-232C



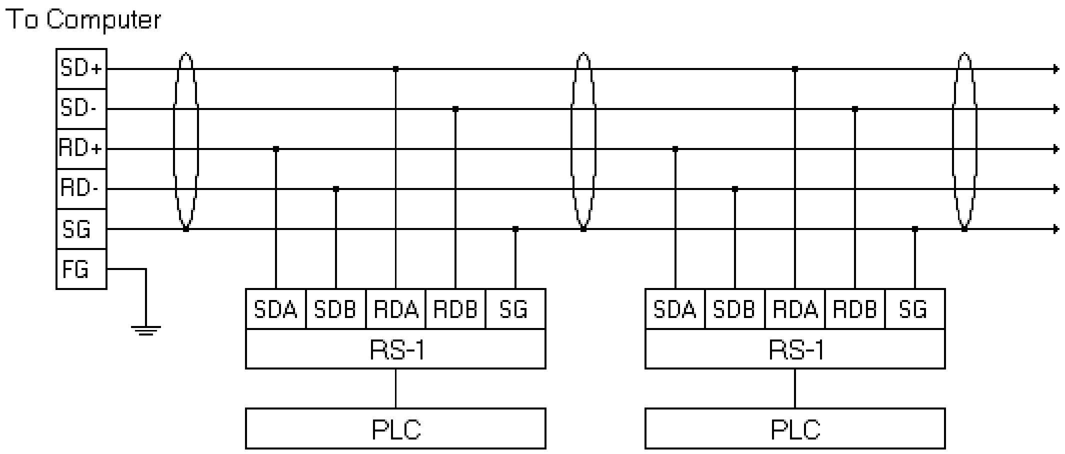
RS-1 RS-232C Interface

Pin	Signal
1	FG
2	SD
3	RD
4	RTS
5	CTS
6	DSR
7	SG
8	CD
20	DTR
22	CI

To Computer

Pin	Signal
1	FG
2	SD
3	RD
4	RTS
5	CTS
6	DSR
7	SG
8	CD
20	DTR
22	CI

1:N Network using RS-485



Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
String	Null terminated ASCII string.

Address Descriptions

Address specifications vary depending on the model in use. Select a link from the following list to obtain specific address information for the model of interest.

Flex-PC NB0
Flex-PC N Series Open

NB0 Model Address Descriptions

The **Flex-PC NB0** model option must be selected if you are communicating with an NB0 device. Use the **Flex-PC N Series Open** model for all other devices. All addresses are in hexadecimal. Bit numbers, string lengths, and array dimensions are in decimal. Default data types are shown in **bold**.

Address Type	Range	Data Type	Access
Input relay (as bits)	X0-X1F	**Boolean**	Read Only
Input relay (as words)*	WX0-WX1 WX0.b-WX1.b (b is bit number 0-15)	**Word**, Short*3 **Boolean**	Read Only
Output relay (as bits)	Y0-Y1F	**Boolean**	Read/Write
Output relay (as words)*	WY0-WY1 WY0.b-WY1.b (b is bit number 0-15)	**Word**, Short*3 **Boolean**	Read/Write
Internal relay (as bits)	M0-M81FF	**Boolean**	Read/Write
Internal relay (as words)*	WM0-WM81F WM0.b-WM81F.b (b is bit number 0-15)	**Word**, Short*3 **Boolean**	Read/Write
Latch relay (as bits)	L0-LFF	**Boolean**	Read/Write
Latch relay (as words)*	WL0-WLF WL0.b-WLF.b (b is bit number 0-15)	**Word**, Short*3 **Boolean**	Read/Write
Timer contact	T0-T1F	**Boolean**	Read/Write
Timer current value	WT0-WT1F	**Word**, Short, BCD*3	Read/Write
Counter contact	C0-C1F	**Boolean**	Read/Write
Counter current value	WC0-WC1F	**Word**, Short, BCD*3	Read/Write
Data register	D0-D803F D0-D803E D0.b-D803F.b (b is bit number 0-15) D0.l-D7FFF.l (l is string length 0-64)	**Word**, Short, BCD*3 DWord, Long, LBCD*2,*3 **Boolean** String*4	Read/Write

*When addressing discrete data as words, addresses correspond to word offsets. For example, WX0 references X0-XF, and WX1 references X10-X1F, etc.

*2 When 32-bit data types are specified, two consecutive 16-bit registers will be used. For example, if D0 is declared as type DWord, registers D0 and D1 will both be used.

*3 These addresses may be also be referenced as arrays. The syntax for declaring an array, using data registers as an example, is: Dxxxx[rows][cols] or Dxxxx[cols] with an assumed row count of 1. For Word, Short and BCD arrays, the base address + (rows*cols) cannot exceed 0x803F. For DWord, Long and LBCD, the base address + (rows*cols*2) cannot exceed 0x803F. In all cases, the total number of bytes being requested cannot exceed the block size. **See Also:** **Block Size**.

*4 ASCII strings can be stored in data registers. When using data registers for string data, each register will contain two bytes of ASCII data. Characters are stored in "low byte to high byte" order. For example, writing the string "ABCD" to D0.4 would result in D0=0x4142 and D1=0x4344.

Open Model Address Descriptions

The **Flex-PC N Series Open** model option allows this driver to handle a wide range of Fuji N series PLCs without specific control over the range of addresses available from the device. If you are using an NB0 device, you must select the **Flex-PC NB0**model. The address ranges shown below may exceed the range available for your particular device. If an address is requested that is not supported by your device, the Fuji Flex driver will mark the requested data item in error. All addresses are in hexadecimal. Bit numbers, string lengths, and array dimensions are in decimal. Default data types are shown in **bold**.

Address Type	Range	Data Type	Access
Input relay (as bits)	X0–XFFFF	**Boolean**	Read Only
Input relay (as words)*	WX0–WXFFFF WX0.b–WXFFFF.b (b is bit number 0–15)	**Word**, Short*3 **Boolean**	Read Only
Output relay (as bits)	Y0–YFFFF	**Boolean**	Read/Write
Output relay (as words)*	WY0–WYFFFF WY0.b–WYFFFF.b (b is bit number 0–15)	**Word**, Short*3 **Boolean**	Read/Write
Internal relay (as bits)	M0–MFFFF	**Boolean**	Read/Write
Internal relay (as words)*	WM0–WMFFFF WM0.b–WMFFFF.b (b is bit number 0–15)	**Word**, Short*3 **Boolean**	Read/Write
Latch relay (as bits)	L0–LFFFF	**Boolean**	Read/Write
Latch relay (as words)*	WL0–WLFFFF WL0.b–WLFFFF.b (b is bit number 0–15)	**Word**, Short*3 **Boolean**	Read/Write
Timer contact	T0–TFFFF	**Boolean**	Read/Write
Timer current value	WT0–WTFFFF	**Word**, Short, BCD*3	Read/Write
Counter contact	C0–CFFFF	**Boolean**	Read/Write
Counter current value	WC0–WCFFFF	**Word**, Short, BCD*3	Read/Write
Step relay	S0–SFFFF	**Boolean**	Read/Write
Data register	D0–DFFFF D0–DFFFE D0.b–DFFFF.b (b is bit number 0–15) D0.l–D7FFF.l (l is string length 0–64)	**Word**, Short, BCD*3 DWord, Long, LBCD*2,*3 **Boolean** String*4	Read/Write

*When addressing discrete data as words, addresses correspond to word offsets. For example, WX0 references X0–XF, and WX1 references X10–X1F, etc.

*2 When 32-bit data types are specified, two consecutive 16-bit registers will be used. For example, if D0 is declared as type DWord, registers D0 and D1 will both be used.

*3 These addresses may be also be referenced as arrays. The syntax for declaring an array, using data registers as an example, is: Dxxxx[rows][cols] or Dxxxx[cols] with an assumed row count of 1. For Word, Short and BCD arrays, the base address + (rows*cols) cannot exceed 0x803F. For DWord, Long and LBCD, the base address + (rows*cols*2) cannot exceed 0x803F. In all cases, the total number of bytes being requested cannot exceed the block size. **See Also:** **Block Size**.

*4 ASCII strings can be stored in data registers. When using data registers for string data, each register will contain two bytes of ASCII data. Characters are stored in "low byte to high byte" order. For example, writing the string "ABCD" to D0.4 would result in D0=0x4142 and D1=0x4344.

Error Descriptions

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation

Missing address
Device address '<address>' contains a syntax error
Address '<address>' is out of range for the specified device or register
Device address '<address>' is not supported by model '<model name>'
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only

Serial Communications

COMn does not exist
Error opening COMn
COMn is in use by another application
Unable to set comm parameters on COMn
Communications error on '<channel name>' [<error mask>]

Device Status Messages

Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Driver Specific Messages

Device '<device name>' returned 'Parameter Error' probably caused by a bad address in block (Tag: '<address>', Size: <size> bytes). Block deactivated
Device '<device name>' returned 'Processing is impossible due to transmission interlock by another device or loader' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned 'Incorrect module number' probably due to unsupported memory type (Tag: '<address>', Size: <size> bytes). Block deactivated
Device '<device name>' returned 'Address exceeding the module's range was specified during write' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned 'Connection to network is impossible' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned 'Another loader is communicating over the network' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned 'Transmission error' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned error code: '<code>' (Tag: '<address>', Size: <size> bytes)
Received a bad check sum (Device: '<device name>', Tag: '<address>', Size: <size> bytes)
Response had unexpected format (Device: '<device name>', Tag: '<address>', Size: <size> bytes)
Response had incorrect data size (Device: `<device name>', Tag: '<tag name>', Size: <size> bytes)

Address Validation

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation

Missing address
Device address '<address>' contains a syntax error
Address '<address>' is out of range for the specified device or register
Device address '<address>' is not supported by model '<model name>'
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only

Missing address

Error Type:

Warning

Possible Cause:

A tag address that has been specified dynamically has no length.

Solution:

Re-enter the address in the client application.

Device address '<address>' contains a syntax error

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically contains one or more invalid characters.

Solution:
Re-enter the address in the client application.

Address '<address>' is out of range for the specified device or register

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically references a location that is beyond the range of supported locations for the device.

Solution:
Verify that the address is correct; if it is not, re-enter it in the client application.

Device address '<address>' is not supported by model '<model name>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically references a location that is valid for the communications protocol but not supported by the target device.

Solution:
Verify that the address is correct; if it is not, re-enter it in the client application. Also verify that the selected model name for the device is correct.

Data Type '<type>' is not valid for device address '<address>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically has been assigned an invalid data type.

Solution:
Modify the requested data type in the client application.

Device address '<address>' is Read Only

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically has a requested access mode that is not compatible with what the device supports for that address.

Solution:
Change the access mode in the client application.

Serial Communications

The following error/warning messages may be generated. Click on the link for a description of the message.

Serial Communications

COMn does not exist

Error Type:
Fatal

Possible Cause:
The specified COM port is not present on the target computer.

Solution:
Verify that the proper COM port has been selected.

Error opening COMn

Error Type:
Fatal

Possible Cause:
The specified COM port could not be opened due an internal hardware or software problem on the target computer.

Solution:
Verify that the COM port is functional and may be accessed by other Windows applications.

COMn is in use by another application

Error Type:
Fatal

Possible Cause:
The serial port assigned to a device is being used by another application.

Solution:
Verify that the correct port has been assigned to the channel.

Unable to set comm parameters on COMn

Error Type:
Fatal

Possible Cause:
The serial parameters for the specified COM port are not valid.

Solution:
Verify the serial parameters and make any necessary changes.

Communications error on '<channel name>' [<error mask>]

Error Type:
Serious

Error Mask Definitions:
B = Hardware break detected.
F = Framing error.
E = I/O error.
O = Character buffer overrun.
R = RX buffer overrun.
P = Received byte parity error.
T = TX buffer full.

Possible Cause:

1. The serial connection between the device and the host PC is bad.
2. The communications parameters for the serial connection are incorrect.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify the specified communications parameters match those of the device.

Device Status Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Device Status Messages

Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Device '<device name>' not responding

Error Type:

Serious

Possible Cause:

1. The serial connection between the device and the host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify the specified communications parameters match those of the device.
3. Verify the Network ID given to the named device matches that of the actual device.
4. Increase the Request Timeout setting so that the entire response can be handled.

Unable to write to '<address>' on device '<device name>'

Error Type:

Serious

Possible Cause:

1. The serial connection between the device and the host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify the specified communications parameters match those of the device.
3. Verify the Network ID given to the named device matches that of the actual device.

Driver Specific Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Driver Specific Messages

Device '<device name>' returned 'Parameter Error' probably caused by a bad address in block (Tag: '<address>', Size: <size> bytes). Block deactivated
Device '<device name>' returned 'Processing is impossible due to transmission interlock by another device or loader' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned 'Incorrect module number' probably due to unsupported memory type (Tag: '<address>', Size: <size> bytes). Block deactivated
Device '<device name>' returned 'Address exceeding the module's range was specified during write' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned 'Connection to network is impossible' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned 'Another loader is communicating over the network' (Tag: '<address>', Size: <size> bytes)

Device '<device name>' returned 'Transmission error' (Tag: '<address>', Size: <size> bytes)
Device '<device name>' returned error code: '<code>' (Tag: '<address>', Size: <size> bytes)
Received a bad check sum (Device: '<device name>', Tag: '<address>', Size: <size> bytes)
Response had unexpected format (Device: '<device name>', Tag: '<address>', Size: <size> bytes)
Response had incorrect data size (Device: `<device name>', Tag: '<tag name>', Size: <size> bytes)

Device '<device name>' returned 'Parameter Error' probably caused by a bad address in block (Tag: '<address>', Size: <size> bytes). Block deactivated

Error Type:
Warning

Possible Cause:
It's likely that an attempt was made to reference a nonexistent location in the specified device.

Solution:
Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations.

Device '<device name>' returned 'Processing is impossible due to transmission interlock by another device or loader' (Tag: '<address>', Size <size> bytes)

Error Type:
Warning

Possible Cause:
An attempt was made to communicate with a device that has already established a 1-to-1 link with another device.

Solution:
Disconnect the offending device from the target device.

Device '<device name>' returned 'Incorrect module number' probably due to unsupported memory type (Tag: '<address>', Size: <size> bytes). Block deactivated

Error Type:
Warning

Possible Cause:
It's likely that an attempt was made to reference a nonexistent location in the specified device.

Solution:
Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations.

Device '<device name>' returned 'Connection to network is impossible' (Tag: '<address>', Size: <size> bytes)

Error Type:
Warning

Possible Cause:
The device could not connect to the network.

Solution:
Verify communications unit settings.

Device '<device name>' returned 'Address exceeding the module's range was specified during write' (Tag: '<address>', Size: <size> bytes)

Error Type:
Warning

Possible Cause:
An attempt was made to reference a nonexistent location in the specified device.

Solution:
Unless the offending tag is not active in the client, the driver will detect this problem during a read and deactivate the tag's block. Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations.

Device '<device name>' returned 'Another loader is communicating over the network' (Tag: '<address>', Size: <size> bytes)

Error Type:
Warning

Possible Cause:
An attempt was made to communicate with a device that has already established a 1-to-1 link with a loader.

Solution:
Disconnect the loader.

Device '<device name>' returned 'Transmission error' (Tag: '<address>', Size: <size> bytes)

Error Type:
Warning

Possible Cause:
A communications error was detected by the device.

Solution:
The driver will automatically retry the request. If error message is frequent, take measures to reduce noise.

See Also:
Device Setup

Device '<device name>' returned error code: '<code>' (Tag: '<address>', Size: <size> bytes)

Error Type:
Warning

Possible Cause:
The device encountered an error while processing a request.

Solution:
Refer to device documentation for meaning of error code and take appropriate actions.

Received a bad check sum (Device: '<device name>', Tag: '<address>', Size: <size> bytes)

Error Type:
Warning

Possible Cause:
The driver detected a communications error.

Solution:
The driver will automatically retry the request. If error message is frequent, take measures to reduce noise.

See Also:
Setup

Response had unexpected format (Device: '<device name>', Tag: '<address>', Size: <size> bytes)

Error Type:

Warning

Possible Cause:

The device issued a response that is not in the format expected by the driver.

Solution:

Contact Technical Support.

Response had incorrect data size (Device: '<device name>', Tag: '<tag name>', Size: <size> bytes)

Error Type:

Warning

Possible Cause:

The device issued a response that did not contain the expected number of data points.

Solution:

It's likely that an attempt was made to reference a nonexistent location in the specified device. Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations.

Index