1782-JDB4 and 1782-JDB8
SmartMux-Lite™ / DeviceNet™
User's Manual

Western Reserve Controls, Inc.

Although every effort has been made to insure the accuracy of this document, all information is subject to change without notice. WRC takes no liability for any errors in this document or for direct, indirect, incidental or consequential damage resulting from the use of this manual.

Document PUB 10.0
Rev 1.10
April 1997

Western Reserve Controls, Inc.
526 South Main Street, Suite 310
Akron OH 44311
330-762-1611 (Phone)
330-762-1622 (FAX)
sales@wrcakron.com (Email)
http://www.wrcakron.com (Web)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Overview

The 1782-JDB4 and 1782-JDB8 SmartMux-Lite™ are remote industrial I/O devices that multiplex discrete data acquisition and control signals in industrial or commercial applications. The 1782-JDB4 and 1782-JDB8 are DIN-mounted I/O multiplexers and support serial communications on a DeviceNet™ link.

The 1782-JDB4 multiplexes four (4) points of discrete I/O and the 1782-JDB8 provides eight (8) points of discrete I/O through WRC's line of digital single point I/O modules, which plug directly into the multiplexer assembly. The 1782-JDB4 and 1782-JDB8 (collectively referred to as "JDBx" or SmartMux-Lite) connect these discrete I/O signals to a DeviceNet communications system. Each channel may be an input or output based upon the type of I/O module selected and each may be any voltage range desired, based upon the module selected. (The term "JDBx" or SmartMux-Lite in this manual refers to either or both of the products when common features or characteristics are discussed. When the information provides refers to only one or the other, that specific product number will be used.)

The SmartMux-Lite multiplexer is designed as a Group 2 Only Server on the DeviceNet system and its I/O is read by and written to by another device on the link defined as a DeviceNet Master (Client). JDBx supports the Predefined Master/Slave Explicit Message Connection, Polled I/O, Change-of-state I/O, Cyclic and Bit-Strobe.

The device address and data rate are software-configurable and are changed from the default by a third-party configuration tool. There are no jumpers and switches for the user to set. Each JDBx has 2 green/red LEDs - one for module status and one for network status. Section 5 of this document describes the installation of the JDBx multiplexer.

WRC also provides SmartMux-Lite products for analog I/O signal conditioning and multiplexing on DeviceNet: the 1782-JDA4 and 1782-JDA8, for 4 points and 8 points, respectively. In addition, WRC also provides products to extend DeviceNet and other CAN based networks. These are the WRC-CANEXT series of CAN bus repeaters.

1.1. DeviceNet System Configuration

A DeviceNet network is a distributed I/O system that may contain many different products from several different vendors. Products may be configured uniformly, as clusters, or as distributed clusters. Up to 64 devices, including the master, may be attached to a single DeviceNet network. Any of these, except the master, may be a SmartMux-Lite. A typical system will include a master, such as a PLC or industrial PC, and multiple slave devices.

1.2. Basic I/O Operation

The JDBx operates as a combined discrete input and discrete output device on the DeviceNet network. It is a slave device which can be allocated by the system implementer to one specific master. There are several parameters that may be configured / modified by the user. These are collected in a block of data called the Parameter Object. (See Section 5.2.) They need not be modified and may be left as defaults, depending upon your application.

The Polled I/O feature follows the "conventional" method of a master requesting data from and/or sending data to one slave at a time. This requires both a command message from the master and a response message from each slave for every set of I/O. To improve throughput on the network Change-of-State, Cyclic I/O and Bit-Strobe functions have been defined by the DeviceNet protocol. These functions are supported by WRC's discrete SmartMux-Lite devices

All data returned to the master will be input data only. There will be a zero (0) value data bit in the response for each channel defined as an output.

1.2.1. Polled I/O

The master can poll (interrogate) the discrete input statuses from the JDBx and energize/de-energize the discrete outputs at the JDBx. The LEDs report the actual state of the input module signals; i.e., they are lighted when the input to the module is energized and extinguished when the input signal is not present. The poll response from the JDBx contains only input data.

Discrete outputs are energized by the master upon a poll command to the discrete output points. The LEDs report the commanded state of the output module signals; i.e., they are lighted when commanded to energize (turn ON) and extinguished when commanded to turn OFF. The status of the outputs are not affected by a poll command with no data (often used to read input data).

The out-of-the-box configuration is that all channels are inputs. Outputs are not enabled by default. Channels that are desired to be outputs must be configured as such. Configuring outputs is done with the the first parameter (Instance 1) of the Parameter Object. See Section 6.2 for the details on how to configure points as outputs.

Both idle and fault operation are implemented for discrete outputs. The outputs can be set individually to hold last state or to implement user-defined states upon receipt of an "idle" command or upon a "fault" condition. You can implement these actions via the Parameter Object.

1.2.2. Change-of-State

A change-of-state (COS) message is an unsolicited messsage containing discrete input data sent by the JDBx to the host wheniver any selected input changes ON-to-OFF or OFF-to-ON. You can individually select no, any or all inputs to cause a COS message. This selection is done by setting the Parameter 2 in the Parameter Object. When an appropriate COS occurs the entire input data (all four or eight input points) are transmitted to the master. Both acknowledged and unacknowledged COS messages are supported. The JDBx will ignore a command to set the COS configuration bit for an output channel and will return an error.

By definition within the DeviceNet spec, a COS will also occur after a specified time interval if no inputs have changed state.

1.2.3. Cyclic I/O

Cyclic I/O is the function by which a slave device sends all its input data to the host at a specific time period without the host explicitly requesting it. (Cyclic I/O is not set up to individually select any or all inputs to cause a cyclic I/O message.) When the specified time interval elapses the entire input data (all four or eight input data) is transmitted to the master.

1.2.4. Bit-Strobe Message

A Bit-Strobe Message is a very fast method by which the master can send one bit to and receive up to 8 bytes of data response from each slave device on the network. The JDBx supports the Bit-Strobe feature and returns its 4 or 8 points (one byte) of input data as a Bit-Strobe Response when a Bit-Strobe Command is received. The JDBx does ignores the one bit sent to it in the command message. No user-defined configuration is required.

2. Using This Manual

This manual serves to help the user to understand the capabilities of the SmartMux-Lite product family, how to install and configure an I/O subsystem using these products, and how to generate the commands from the system host to read data from or write data /to the SmartMux-Lite.

Section 3 describes how to quickly connect your SmartMux-Lite and get it up and running on the DeviceNet link.

Section 4 provides the technical specifications for the products described in this manual.

Section 5 describes the installation of the hardware, including mounting, connection to other I/O subsystem components, power requirements and configuration of the operating parameters of the SmartMux-Lite.

Section 6 describes the DeviceNet Profile and DeviceNet Object data for these products.

Section 7 lists common accessories that are used with the JDB4 or JDB8..

Section 8 provides some troubleshooting hints in the event your SmartMux-Lite or I/O system is not operating as anticipated.

3. Quick Start

To quickly and easily install your SmartMux-Lite in your DeviceNet system, follow the instructions below. For more details, see Section 5.

To Install and DeviceNet Establish Communications

1. Remove the SmartMux-Lite from the box and connect your DeviceNet cable to the 5-pin plug (supplied) according to DeviceNet cable wiring specifications.
2. Make sure that the DeviceNet network is terminated properly.
3. Make sure that there is power on the DeviceNet network and plug the cable into the SmartMux-Lite.
4. The SmartMux-Lite will undergo its initialization sequence, flashing both LEDs. After approximately 5 seconds, the Module Status LED (labeled "MS") will go on solid green and network LED will flash green.
5. The green Network Status LED (labeled "NS") will go on solid once the Master recognizes the unit on the link and allocates the connection.
6. The SmartMux-Lite is now operating on the network.

To Configure the Node Address and Baud Rate

1. You may change the device address and/or the baud rate, using your configuration tool or any command from the Master. (Defaults are 63 and 125k baud.)
2. If you change the device address, the SmartMux-Lite will reset and assume the address.
3. If you change the baud rate, the new baud rate will not become effective until the unit is power cycled or a Reset command is received from the Master.

To Read Discrete Input Data

1. Allocate a Poll Connection (Connection Instance 2) to the SmartMux-Lite from the Master.
2. Perform a poll command to the SmartMux-Lite from the Master. The input channel values will be available in bits 0-3 or 0-7 (for JDB4 or JDB8) of the one data byte returned.

To Define a Channel as a Discrete Output

1. Set **Parameter Instance 1, Attribute 1** to a value of 0-F_{hex} or 00-FF_{hex} . Each bit set to 1 represents a corresponding discrete channel to be defined as an output.

   Note: Changing the Discrete Output configuration will cause any previously defined outputs to be turned **off**.

To Energize and De-energize Discrete Outputs

1. Issue a Poll command from the Master with a data value of 00-FF $_{hex}$. Every defined output will be turned ON or OFF, as defined by a corresponding bit value of 1 or 0.
2. Output data sent to channels not defined as outputs will be ignored for those channels only. Output commands to other valid output channels will be executed as anticipated.

To Re-define an Output Channel as a Discrete Input

1. Set **Parameter Instance 1, Attribute 1** to a value of 0-F $_{hex}$ or 00-FF $_{hex}$. Each bit set to 0 represents a corresponding discrete channel to be re-defined as an input.

To Define Input Channels as Change-of-State (COS) Inputs and Activate COS Operation

1. Set **Parameter Instance 2, Attribute 1** to a value of 0-F $_{hex}$ or 00-FF $_{hex}$. Each bit set to 0 represents a corresponding discrete channel which will cause the JDBx to generate a COS message.
2. See your configuration tool documentation for the specific instructions required to perform the following functions.
3. Make sure that the Cyclic I/O connection is not allocated.
4. Using your configuration tool, Allocate the COS function. That is: Allocate Connection Class 4 -- using the following values:

   10 (hex) for Acknowledged COS

   50 (hex) for Unacknowledged COS
5. Set the EPR (expected packet rate) for the maximum time interval for a COS message. Set EPR, Connection Instance 4 to xxx milliseconds.
6. To re-define the specific COS inputs after Steps 2 - 4 have been performed, only Step 1 needs to be repeated.

To Activate Cyclic I/O Operation

1. See your configuration tool documentation for the specific instructions required to perform the following functions.
2. Make sure that the COS I/O connection is not allocated.
3. Using your configuration tool, Allocate the Cyclic I/O function. That is: Allocate Connection Class 5 -- using the following values:

   20 (hex) for Acknowledged Cyclic I/O

   60 (hex) for Unacknowledged Cyclic I/O
4. Set the EPR (expected packet rate) for the maximum time interval for a COS message. Set EPR, Connection Instance 4 to xxx milliseconds.
5. To re-define the specific COS inputs after Steps 2 - 4 have been performed, only Step 1 needs to be repeated.

4. General Specifications

Product: 1782-JDB4 / 1782-JDB8 SmartMux-Lite DeviceNet for Single-Point Discrete I/O Modules

Description: Remote multiplexer, compatible with ODVA's DeviceNet protocol for discrete I/O, which will directly support up to four (4) / eight (8) WRC 1781 series discrete I/O modules

Device Profile: General Purpose Discrete I/O, Class $0x07_{hex}$
plus Vendor-Specific Discrete Output (Class 64_{hex})

Product Revision: 1.10

DeviceNet Conformance: Designed to conform to the ODVA DeviceNet Specification Volume I, Version 1.3 and Volume II, Version 1.2.

Communications: Predefined Master/Slave Connection Set, Group 2 Only Server

Baud rate: 125k, 250, 500k - software selectable (default = 125k)

Address selection: Address number 1 to 63, software selectable (default = 63)

DeviceNet Connection:

SmartMux-Lite:	5-pin pluggable header (male) Phoenix Contact MSTBA 2.5/5-G-5.08/AU
DeviceNet Cable:	5-contact plug (female contacts) Phoenix Contact MSTB 2.5/5-ST-5.08/AU (included)

Status Indicators:

Module Status:	green/red bi-color LED
Network Status:	green/red bi-color LED

Voltage Isolation: Defined and provided by I/O modules (typically 4000V)

Maximum power: 200 mA @ 11 Vdc - 90 mA @ 25V dc unregulated power supply, including I/O modules (power provided by the DeviceNet network)

Discrete I/O: up to 4 / 8 discrete modules
refresh rate <500 usec for all inputs and outputs

I/O Wiring: One / two 8-pole open, pluggable connector: PCD ELFP08110 or equivalent
I/O wire size: stranded 12 - 26 AWG
solid 14 - 26 AWG

I/O Fuses: One 5A fast-blow Pico II fuse per I/O point provided, Littelfuse number 251005

I/O LEDs: One LED per I/O point - defines microprocessor status of input or output

Mounting: DIN rail mount, EN 50022

Size:
Width: 2.80" (71,0 mm) - JDB4
4.17" (106,0 mm) - JDB8
Height: 3.56" (90,4 mm)
Depth: 1.75" (44,4 mm) - without modules, excluding DIN rail
2.69" (68,30 mm) - with modules, excluding DIN rail

Operating Temp: 0-70 ºC

Humidity: 0-95% RH, non-condensing

5. Hardware Installation and Configuration

5.1. Overview

A SmartMux-Lite discrete I/O subsystem consists of a JDBx multiplexer and one or more WRC 1781-Series discrete I/O modules. The SmartMux-Lite is to be mounted on a EN50022 DIN rail (available from WRC and WRC's distributors as part number WRC 50022) in any orientation. The I/O module(s) plug into the top surface of the SmartMux-Lite and are secured to the device by the captive hold-down screw in each module.

Any position (0-3 for a JDB4 or 0-7 for a JDB8) can be any WRC 1781-Series discrete single-point module. Each position can be input or output, any voltage or signal level, or not-used. Each module position has an LED to indicate the status of the signal.

The JDBx contains two LEDs to indicate the status of the device and the status of the network. The device can be connected to the main DeviceNet trunk line or to a drop line via screw terminations on the open, pluggable 5-pin DeviceNet connector supplied with your unit.

Field wiring to the I/O modules is also provided by screw terminations on pluggable 8-pole connectors. These are supplied with your unit.

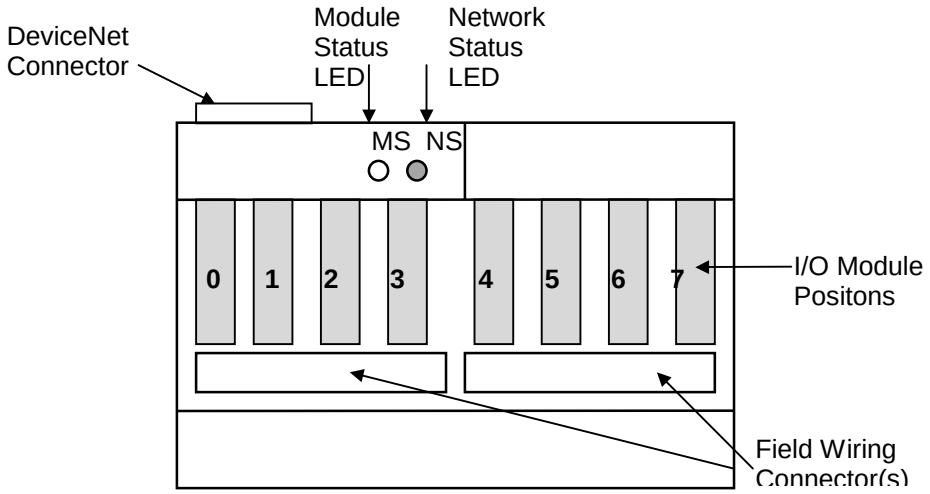

Figure 5-1 1782-JDB8 SmartMux-Lite

5.2. Configuration

A JDBx Multiplexer is software-configured for several parameters. The table below defines the legal values and the default values for the device address (refered to as the device's MacID), baud rate and I/O configuration selections available.

Table 5-1 Configuration Parameters

Parameter	Param. Instance	Parameter Choices	Default Setting	Default Value
Outputs Enabled	1	Each Channel Disabled/Enabled	No Outputs Enabled	0
Change-of-State	2	Each Channel Disabled/Enabled	No COS Inputs Active	0
Output Fault Action	3	Hold Last State/Value	Fault Value	0
Output Fault Value	4	OFF/ON	Turn OFF	0
Output Idle Action	5	Hold Last State/Value	Idle Value	0
Output Idle Value	6	OFF/ON	Turn OFF	0

Definitions of these parameters are as follows:

1. **Outputs Enabled**: A bit for each module position which determines whether the position will use output modules or input modules.
2. **Change-of-State**: A bit for each input position which will determine whether or not a changes in the value to that position will result in all the input data being sent immediately to the JDBx's master device.
3. **Output Fault Action**: Selection to determine whether each output will hold its last state or assume the value identified in the next parameter upon a device fault.
4. **Output Fault Value**: The value each output will assume after a Fault if hold last state is not selected.
5. **Output Idle Action**: Selection to determine whether each output will hold its last state or assume the value identified in the next parameter if an Idle Command is issued by the Master.
6. **Output Idle Value**: The value each output will assume upon an Idle Command if hold last state is not selected.

5.3. LED Operation

A JDBx Multiplexer has two LEDs that provide visual status information to the user about the product and the DeviceNet network. See Figure 5-1, Table 5-2 and Table 5-3.

Table 5-2 Module Status LED (labeled MS)

LED State	Module Status	Meaning
OFF	No Power	There is no power through DeviceNet.
Green	Device Operational	JDBx is operating normally.
Flashing Green	Device in Standby	JDBx needs commissioning.
Flashing Red	Minor Fault	Recoverable fault.
Red	Unrecoverable Fault	JDBx may need replaced.
Flashing Red/Green	Device Self-Testing	JDBx is in self-test mode.

Table 5-3 Network Status LED (labeled NS)

LED State	Module Status	Meaning
OFF	No Power / Not on-line	JDBx has no power or has not completed the Dup_MAC_ID test.
Flashing Green	On-line, not connected	JDBx is on-line but is not allocated to a Master.
Green	On-line	JDBx is operating normally.
Flashing Red	Connection time-out	One or more I/O connections are timed out.
Red	Critical link failure	JDBx has detected an error which makes it incapable of communicating on the link. (Bus off or Duplicate MAC ID).

5.4. Discrete I/O

Table 5-4 lists the selection of discrete I/O modules that can be used with the JDBx multiplexers. These modules provide signal-to-signal and signal-to-system isolation up to 4000 volts. Refer to WRC's full line catalog for detailed I/O module information.

Table 5-4 1781 Series Discrete I/O Modules

1781-	Input / Output	Voltage Type	24 Vdc Power Requirements	24 Vdc Current Requirements
IA5S	Input	AC/DC	65 mW	2.6 mA
IB5S	Input	DC	65 mW	2.6 mA
IM5S	Input	AC/DC	65 mW	2.6 mA
IN5S	Input	AC/DC	65 mW	2.6 mA
IT5S	Input	DC	65 mW	2.6 mA
OA5S	Output	AC	65 mW	2.6 mA
OB5S	Output	DC	65 mW	2.6 mA
OC5S	Output	DC	65 mW	2.6 mA
OF5S	Output	DC	65 mW	2.6 mA
OM5S	Output	AC	65 mW	2.6 mA
RO5S	Output	AC/DC/analog	260 mW	10.4 mA
RC5S	Output	AC/DC/analog	260 mW	10.4 mA
WO5S	Output	AC/DC/analog	260 mW	10.4 mA
WC5S	Output	AC/DC/analog	260 mW	10.4 mA

Warning: Do not insert and remove the I/O modules under power. Doing so may result in unexpected events in your operation and/or damage to the unit.

Note: Each I/O point is provided with a replaceable 5 Amp quick acting fuse. Replacements are available as 1781-FUSE5 from WRC and WRC distributors.

5.4.1. Field I/O Wiring

The field I/O is connected to the JDBx via pluggable ("quick disconnect") connectors, supplied with the device. The JDB4 has one and the JDB8 has two eight-position terminal block(s) and mating connector(s). The mating connector is PCD ELFP08110. (Use this or equivalent component for spares or replacements.) (Available as a spare part from WRC or WRC distributor's as 1781-JD-CK) This allows the user the convenience of wiring the I/O to the connector at a location remote to the actual SmartMux-Lite. It is also convenient during commissioning, maintenance and trouble-shooting of the field wiring. Table 5-5 defines which terminations are used for each I/O channel.

Table 5-5 Field I/O Wiring Connections

Channel No.	Positive Terminal	Negative Terminal
0	2	1
1	4	3
2	6	5
3	8	7
4	10	9
5	12	11
6	14	13
7	16	15

Important: All dc output modules and some dc input modules are polarized. Proper operation requires that they be wired per the above chart. They will not work and may be damaged if the field wiring does not conform to the above chart.

The pluggable wiring connector can accept I/O wire gage sizes 12 to 22 AWG.

Each channel of I/O is individually protected with 5 Amp fast-blow Pico fuses.

5.5. Power Requirements

The JDBx SmartMux-Lite subsystem is powered from the 11-25 Vdc provided by the DeviceNet network. The JDBx consumes 45 mA of current at 24 Vdc, or 1.1 Watts, typical. Table 5-4 lists the power requirements of the discrete I/O modules used with the JDBx. To determine the total power required by the SmartMux-Lite, be sure to use is the total of the JDBx and the I/O modules selected. The maximums are given in Table 5-6 below.

Table 5-6 JDBx Power Requirements

Product	Max Power (no Relay Modules)	Max Power (with Relay Modules)
JDB4	1.4 W	2.1 W
JDB8	1.7 W	3.2 W

Power to and from the field actuators and sensors connected to the I/O modules is supplied by the user from the field wiring. No other external power supply is required to operate the JDBx.

5.6. Network Configuration

DeviceNet specifications provide for a maximum network distances for the main trunk line and drop lines, depending upon the baud rate used on the network. They are:

Table 5-7 Network Maximum Lengths

Baud Rate	Trunk Line Length		Drop Length			
	Maximum Distance		Maximum		Cumulative	
	Meters	Feet	Meters	Feet	Meters	Feet
125k baud	500 m	1640 ft	6 m	20 ft	156 m	512 ft.
250k baud	250 m	820 ft	6 m	20 ft	78 m	256 ft.
500k baud	100 m	328 ft	6 m	20 ft	39 m	128 ft.

5.6.1. Network Termination

A DeviceNet system **must be terminated at each end of the trunk line**. The host controller and the **last** SmartMux-Lite or other DeviceNet device on the network must always be terminated to eliminate reflections, even if only two nodes are present. The DeviceNet specifications for the terminating resistor are:

- 121 ohm
- 1% metal film
- 1/4 Watt

An appropriate terminating resistor kit, WRC part number RM121DN, is included with your SmartMux-Lite Manual - attached below.

Important: Per the DeviceNet spec -- do not terminate devices on drop lines.

5.6.2. DeviceNet Connection Wiring

The supplied DeviceNet connection plug accepts cable sizes from 12 AWG - 24 AWG. The maximum wire size (12 AWG) has an area of 6530 circular mils and the smallest (24 AWG) has an area of 3265 circular mils.

Where not prohibited by local government or wiring regulations or company policy, multiple wires can be inserted each connection point on the plug as long as the total wire area does not exceed that of a 12 AWG wire. Use the chart below as a guide.

Phoenix Contact recommends using the same size wires be used when connecting more than one wire in a screw termination.

UL may require the use of crimped ferrules to connect multiple wires together.

Table 5-8 DeviceNet Wiring Termination

Wire AWG	Wire area (circular mils)	Maximum Wires per Terminal
12	6530	1
14	4110	1
15	3265	2
16	2580	2
18	1620	4 *
20	1022	6 *
22	645	9 *
24	404	16 *

* WRC does not recommend using more than 2 wires in any wire terminal.

The conductor sizes for DeviceNet cables are:

Table 5-9 DeviceNet Conductor Sizes

Function	Thick Wire	Thin Wire
Power	15 AWG	22 AWG
Signal	18 AWG	24 AWG

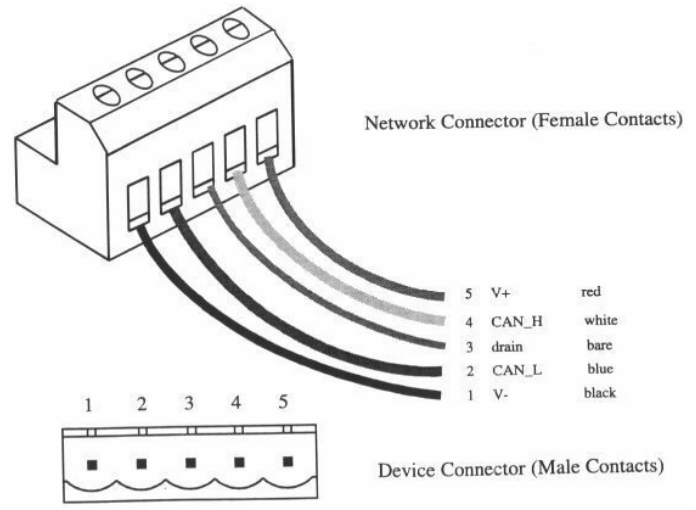

Figure 5-2 DeviceNet cable connector

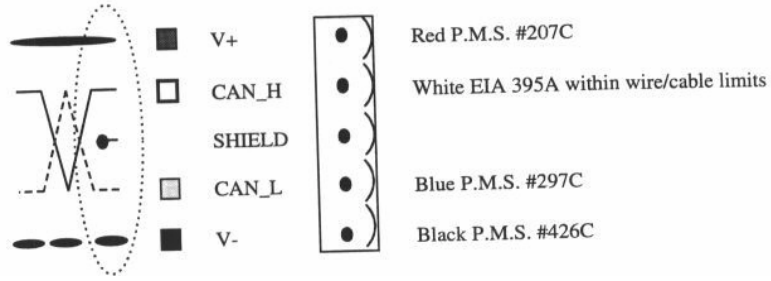

Figure 5-3 DeviceNet cable specifications

6. DeviceNet Profile and DeviceNet Objects

This section describes the DeviceNet Objects present in the SmartMux-Lite. The SmartMux-Lite conforms to a Type 7, General Purpose Discrete I/O Device, with an additional Vendor-Specific Object, the JDB Discrete Output Assembly, added.

Table 6-1 JDB4/JDB8 Device Profile

Object	# of Instances
Identity	1
Message Router	1
DeviceNet	1
Connection	4 (Explicit Msg, Polled I/O, COS/Cyclic, Bit-Strobe)
Parameter	1
Assembly	2 (Input, Output)
Discrete Input	4 / 8
JDBx Discrete Output	4 / 8
Acknowledgment Handler	1

The Discrete Input Assembly Object is defined in Section 6.1 and the JDBx Discrete Output Assembly Object (Vendor-specific) is defined in Section 6.2.

6.1. Discrete Input Assembly - Class 08 $_{hex}$

The input data from the modules plugged into the SmartMux-Lite are returned to the Master as a response to a Poll command in the fashion shown in The JDB4 uses DeviceNet Discrete Input Assembly Instance 3, which means that the data returned is one byte long and is exactly four bits and the JDB8 uses Discrete Input Assembly Instance 4.

Each bit of Data Byte 0 represents a physical module position on the SmartMux-Lite. For each module position that contains an input module, the status of the field device connected to that module will be given in its corresponding position, as a 0 or 1. For each module position that is empty or contains an output module, a value of 0 (zero) will be returned in the corresponding bit location.

Table 6-2 and Table 6-3. The input data received by the Master from the JDBx are contained in the first (and only) byte of data.

6.1.1. Discrete Input Assembly Data Format

The JDB4 uses DeviceNet Discrete Input Assembly Instance 3, which means that the data returned is one byte long and is exactly four bits and the JDB8 uses Discrete Input Assembly Instance 4.

Each bit of Data Byte 0 represents a physical module position on the SmartMux-Lite. For each module position that contains an input module, the status of the field device connected to that module will

be given in its corresponding position, as a 0 or 1. For each module position that is empty or contains an output module, a value of 0 (zero) will be returned in the corresponding bit location.

Table 6-2 JDB4 Discrete Input Assembly Data Attribute Format (Instance 3)

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved				Input 3	Input 2	Input 1	Input 0

Table 6-3 JDB8 Discrete Input Assembly Data Attribute Format (Instance 4)

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Input 0

Example:

Consider a JDB8 with the following configuration:

Channel 7	Channel 6	Channel 5	Channel 4	Channel 3	Channel 2	Channel 1	Channel 0
Input ON	Input OFF	Output ON	Output OFF	Input ON	Empty	Input ON	Output ON

the poll command will return the binary pattern 10001010, or 8A(hex).

6.1.2. Attributes and Services

Following are the Class Attributes, Instance Attributes and Services are supported by the SmartMux-Lite for the Discrete Input Point Object.

Table 6-4 Class 08_{hex} Class Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this is two (2).
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the JDBx.	The current value assigned to this is three (3) / four (4).
6	Get	Max ID Number of Class Attributes	UINT	The attribute ID number of the last class attribute of the class definition implemented in the JDBx.	The current value assigned to this is seven (7).
7	Get	Max ID Number of Instance Attributes	UINT	The attribute ID number of the last instance attribute of the class definition implemented in the JDBx.	The current value assigned to this is three (3).

Table 6-5 Class 08 $_{hex}$ Instance Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of Value
3	Get	Value	BOOL	Input point value	0 = OFF 1 = ON

Table 6-6 Discrete Input Common Services

Service Code	Class	Instance	Service Name	Description of Service
OE $_{hex}$	Get	Get	Get_Attribute_Single	Returns the contents of the specified attribute.

6.2. Discrete Output Point Object - Class 64 $_{hex}$

The JDB4 uses Discrete Output Assembly Instance 33 and the JDB8 uses Discrete Output Assembly Instance 34. The output data to energize or de-energize the modules and connected field devices is sent from the Master with a Poll command in the fashion shown.

Each bit represents a physical module position on the SmartMux-Lite. For each module position that contains an output module, the state of the field device connected to that module will be commanded by the 1 or 0 (one or zero) in its corresponding bit position in the data field. For each module position that is empty or contains an input module, any data sent to that location will be ignored by the SmartMux-Lite.

Before any output can be energized, you must first configure that I/O channel to be enabled as an output.

6.2.1. Discrete Output Assembly Data Format

Table 6-7 JDB4 Discrete Object Assembly Instance Data Format (Instance 33)

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved				Output 3	Output 2	Output 1	Output 0

Table 6-8 JDB8 Discrete Object Assembly Instance Data Format (Instance 34)

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1	Output 0

The WRC JDB Discrete Output Point models the Discrete Output Point (DOP) Model of the DeviceNet Spec, Volume II, Release 1.2, with one exception: it does **not** go to the Idle State upon receipt of a Poll Command with no application data. (The poll command with no data reads discrete input data.)

6.2.2. Attributes and Services

Following are the Class Attributes, Instance Attributes and Services are supported by the SmartMux-Lite for the Discrete Output Point Object.

Table 6-9 Class 64_{hex} Class Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this is one (1).
2	Get	Max. Instance	UINT	Maximum instance number of an object currently	The current value assigned to this is thirty-three (33) / thirty-four (34).
6	Get	Max. ID Number of Class Attribute	UINT	Attribute ID number of the last class attribute of the class definition implemented in the device	The current value assigned to this is seven (7).
7	Get	Max. ID Number of Instance Attributes	UINT	Attribute ID number of the last instance attribute of the class definition implemented in the device	The current value assigned to this is nine (9).

Table 6-10 Class 64_{hex} Instance Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of Value
1	Get	Number of Attributes	USINT	Number of attributes supported in this product	
2	Get	Attribute list	ARRAY OF USINT	List of attributes supported in this product	
3	Set	Value	BOOL	Output point value	0 = OFF 1 = ON
4	Get/Set	Enable	BOOL	Defines the output as enabled or disabled	0 = not active / output disabled 1 = Active / output enabled
5	Set	Fault Action	BOOL	State of output after recoverable failure	0 = Fault Value attribute 1 = Hold Last State
6	Set	Fault Value	BOOL	User-defined value for use with Fault State attribute	0 = OFF 1 = ON
7	Set	Idle Action	BOOL	State of output during idle	0 = Idle Value attribute 1 = Hold Last State
8	Set	Idle Value	BOOL	User-defined value for use with Idle State attribute	0 = OFF 1 = ON
9	Set	Run_Idle_Command	BOOL	Changes state of Digital Output to Idle Mode or Run Mode	0 = IDLE 1 = RUN

Table 6-11 Discrete Output Common Services

Service Code	Class	Instance	Service Name	Description of Service
OE_{hex}	Get	Get	Get_Attribute_Single	Returns the contents of the specified attribute.
10_{hex}	n/a	Set	Set_Attribute_Single	Modifies an attribute value.

6.3. Parameter Object - Class $0F_{hex}$

There are many configurable data parameters associated with your SmartMux-Lite. The JDBx uses a Parameter Object (a collection of these parameters) to assist you in reading and changing configurable data. The parameter list is given in Table 5-1 in Section 5.2.

6.3.1. Attributes and Services

Following are the Class Attributes, Instance Attributes and Services are supported by the SmartMux-Lite for the Parameter Object.

Table 6-12 Class $0F_{hex}$ Class Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of Value
2	Get	Max. Instance	UINT	Maximum instance number of an object currently	The current value assigned to this is eight (8).
8	Get	Max. ID Number of Class Attribute	UINT	Attribute ID number of the last class attribute of the class definition implemented in the device	The current value assigned to this is twelve (12).
9	Get	Configuration Assembly Instance	UINT	Instance Number of the Configuration Assembly.	The current value assigned to this is zero (0. No configuration assembly is provided.
10	Set	Native Language	UINT	Language ID for all character array accesses.	The current value assigned to this is zero (0) -- English.

Table 6-13 Class 0F $_{hex}$ Instance Attributes

Attribute ID	Access Rule	Stub/ Full	Name	DeviceNet Data Type	Description of Attribute	Semantics of Value
1	Set		Parameter Value	*data type* specified in Descriptor Data Type and Data Size.	Actual value of parameter. It can be read from or written to. This attribute is read-only if bit 4 of Attribute 4 is TRUE.	
2	Set		Link Path Size	USINT	Size of link path. If this is 0, then no link is specified.	Number of bytes.
3	Set		Link Path	ARRAY of DeviceNet path:	DeviceNet path to the object from where this parameter's value is retrieved.	
4	Get		Descriptor	WORD	Description of parameter.	
5	Get		Data Type	USINT	Data type code.	
6	Set		Data Size	USINT	Number of bytes in Parameter Value	

Table 6-14 Parameter Common Services

Service Code	Class	Instance	Service Name	Description of Service
OE $_{hex}$	Get	Get	Get_Attribute_Single	Returns the contents of the specified attribute.
10 $_{hex}$	n/a	Set	Set_Attribute_Single	Modifies an attribute value.

6.4. Acknowledge Handler Object - Class 2B $_{hex}$

The DeviceNet specification requires the use of both acknowledged and un-acknowledged change-of-state (COS) messages The JDBx uses the DeviceNet formal Acknowledge Handler to allow the master to request and receive either acknowledged or un-acknowledged change-of-state (COS) messages. In general the use of acknowledged COS provides for the slave device to take action if, for some reason, the master does not acknowledge the COS after a certain number of retries. This occurs, however, at the expense of increased network traffic. To reduce this traffic and increase network efficiency, unacknowledged COS messages may be desired.

If the JDBx is operating in acknowledged mode, it will try to send the same COS message a second time if the first is not acknowledged. If still no acknowledgement is received, then no further action is taken and the JDBx continues to operate as normal.

6.4.1. Attributes and Services

Following are the Class Attributes, Instance Attributes and Services are supported by the SmartMux-Lite for the Acknowledge Handler Object.

Table 6-15 Class 2B $_{hex}$ Class Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Description of Attribute	Semantics of Value
1	Get	Revision	UINT	Revision of this object.	The current value is 1.
2	Get	Max Instance	UINT	Maximum instance number of an object currently	The current value assigned to this is ().

Table 6-16 Class 2B $_{hex}$ Instance Attributes

Attribute ID	Access Rule	Stub/ Full	Name	DeviceNet Data Type	Description of Attribute	Semantics of Value
1	Set		Acknowledge Timer	UINT	Time to wait for acknowledge before resending.	Range 1-65,535. Default = 16. (0 is invalid.)
2	Get		Retry Limits	USINT	Number of Ack Time-outs to wait before informing producing application of a Retry_Limit_Event reached.	Range 0-255. Default is 1.
3	Get		COS Producing Instance	UINT	Connection Instance which contains the path of the producing I/O application object which will be notified of Ack Handler events.	Connection Instance ID. Current setting is 4.

Table 6-17 Acknowledge Handler Common Services

Service Code	Class	Instance	Service Name	Description of Service
OE $_{hex}$	Get	Get	Get_Attribute_Single	Returns the contents of the specified attribute.
10 $_{hex}$	n/a	Set	Set_Attribute_Single	Modifies an attribute value.

7. Accessories

The following components can be used with a 1782-JDBx SmartMux-Lite for replacements or spare parts.

Part	WRC Part Number
• I/O connectors for field wires	1782-JDB-CK
• DeviceNet link connector plug	1782-JD-DN-CK
• I/O Module Fuses (10 piece kit)	1781-FUSE5
• DIN rail	WRC 50022

8. Troubleshooting

This section identifies some of the common problem observed when commissioning or operating a DeviceNet and SmartMux-Lite.

Problem:

Module Status LED is solid Green
Network Status LED is flashing Green
Device will not communicate on the network

Possible Solutions:

1. Network does not have a terminating resistor. Add a 121 ohm resistor across the CAN_H and CAN_L signals at the first and last nodes.

Problem:

Module Status LED is solid Red
Network Status LED is flashing Green or solid Green
Device will not perform I/O

Possible Solutions:

1. One or more channels were configured as Discrete Outputs and an input module with an active input signal is located in one or more of those channel positions.

Problem:

An output device does not turn on when commanded. Module LED is OFF.

Possible Solutions:

1. An output module is not plugged into the channel selected.
2. The selected channel was not configured as an output.
3. The unit is damaged.

Problem:

An output device does not turn on when commanded. Module LED is ON.

Possible Solutions:

1. The fuse is blown.
2. There is a loose connection in the field wiring.
3. The field wiring is not properly connected from hot to the module to the actuator to common.
4. If it is a dc module, the wiring to the module is the wrong polarity.
5. The module is damaged.