MYNAH℠

ECMA-24 Driver
Programmable Serial Interface Card

USER MANUAL

Rev. P1.55

March 2010

Printed in the U.S.A.

1 INTRODUCTION

1.1 Scope

This document is the User Manual for the ECMA-24 serial communication driver firmware for the Emerson Process Management (EPM) DeltaV Control System; it provides information required to install, configure, and maintain the driver firmware on the DeltaV Programmable Serial Interface Card (PSIC). The reader should be familiar with EPM's DeltaV PSIC and connected field devices supporting the ECMA-24 protocol.

The section *Document Format* briefly describes the contents of each section of this manual. *System Specifications* outlines hardware and software requirements for the ECMA-24 Driver firmware.

1.2 Document Format

This document is organized as follows:

Introduction	Describes the scope and purpose of this document.
Theory of Operation	Provides a general functional overview of the ECMA-24 Driver.
Flashing Firmware	Describes flashing procedures for the ECMA-24 Driver firmware on to the DeltaV PSIC.
Configuration Information	Describes procedures and guidelines for configuring the DeltaV PSIC.
Operational Check	Provides tips and assistance to ensure PSIC is properly setup and configured.
DeltaV–Field Device Electrical Interface	Describes the electrical interface between DeltaV PSIC and the ECMA-24 Device. Also describes the cable pin assignments for RS-232 and RS-422/485 communications.
Technical Support	Describes who to call if you need assistance.

1.3 System Specifications

The following table lists the minimum system requirements for the ECMA-24 Driver:

Table 1: System Specifications

Firmware	ECMA-24 Driver Firmware
Protocol Compatibility	• TCS Communications ECMA 24 Protocol, Revision C, Daniel Measurement and Control, 23 July 1999. • ECMA 24 Design, AMOCO EkoFisk, Rev 0, 17 March 1997. • Standard ECMA-24 for Code Independent Information Transfer, December 1969.
Software Requirements	DeltaV System Software (Release 4.2 or later) installed on a hardware-appropriate Windows workstation configured as a ProfessionalPlus for DeltaV Serial Interface Port License (VE4102) if required.
Minimum DeltaV Hardware Requirements	DeltaV Series 2 Serial Module, Hardware Rev 1.1r or later DeltaV M3, M5, M5+, MD, MD Plus or MX Controller, Power Supply and 8 wide controller carrier

2 THEORY OF OPERATION

DeltaV comprises an I/O sub-system, in which the PSIC is one type of card. The purpose of the PSIC is to serially integrate third-party devices, allowing data to be read into and written out from DeltaV. Each PSIC has 2 communication ports that can be configured as Master or Slave, using RS-232, RS-485 (Half Duplex), or RS-422 (Full Duplex). Various communications parameters, such as baud rate, are configurable.

For the ECMA-24 application, the PSIC driver functions only in Slave mode, while the external device functions as the communications Master.

On power-up, the PSIC driver receives its configuration from the DeltaV Controller. The driver then waits for and responds to master device messages to send/receive data.

All bi-directional data transfers are initiated by the master device. The data received from the master device is communicated back to the DeltaV Controller. The PSIC sends DeltaV data to the master device as a user driven command. Specifically, the user command simply marks the block data as available for transmission. A subsequent invitation to send from the master initiates the data transfer, and the marked block is cleared on successful completion of send. A second invitation to send results in a “no new data” message response.

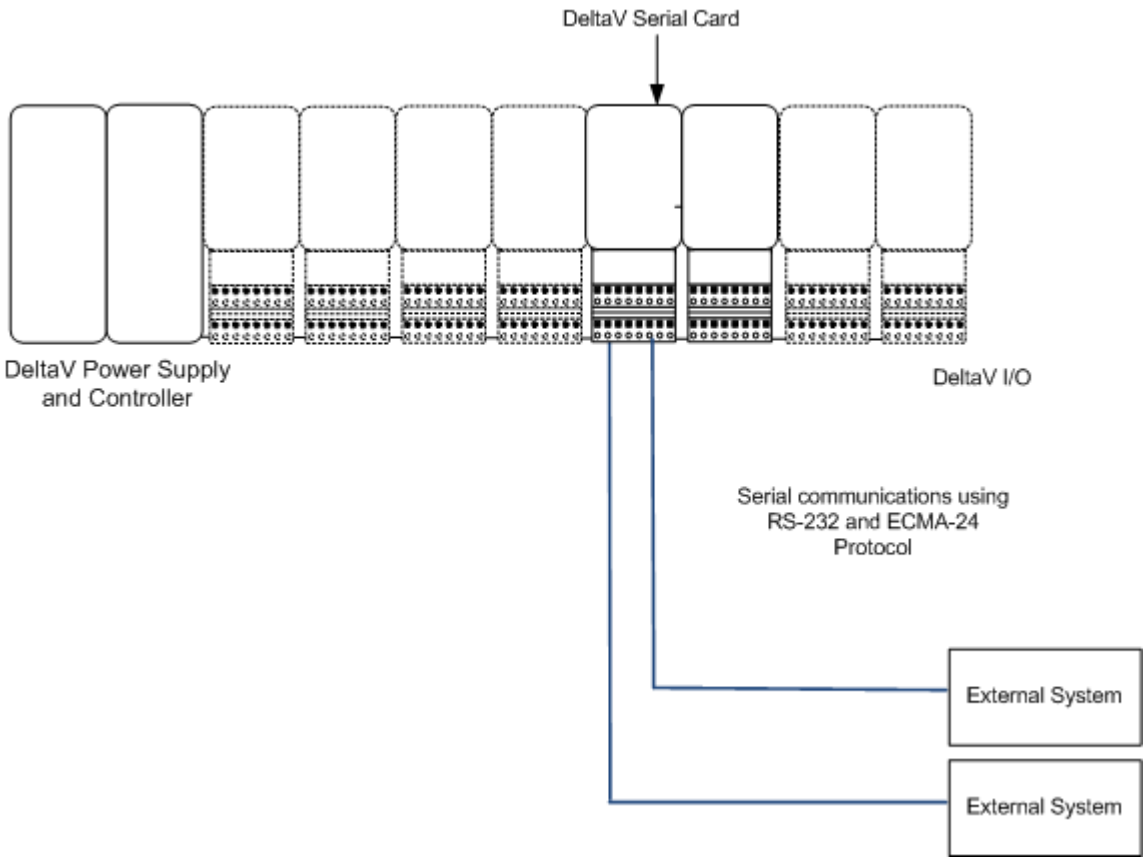
The PSIC configuration entered by users comprises the following:

1. Only one device with specific address under each port.

2. Maximum of 15 datasets under the device, configured as five (5), three (3) dataset groups. Each dataset group comprises parameters to configure three (3) data block structures and to hold dynamic data. The dataset configuration parameters uniquely identify the blocks to be read or written.

3. The driver functions as a Slave. The master initiates all communications and the driver uses the configured data blocks to send/receive data.

The physical architecture will be as follows:

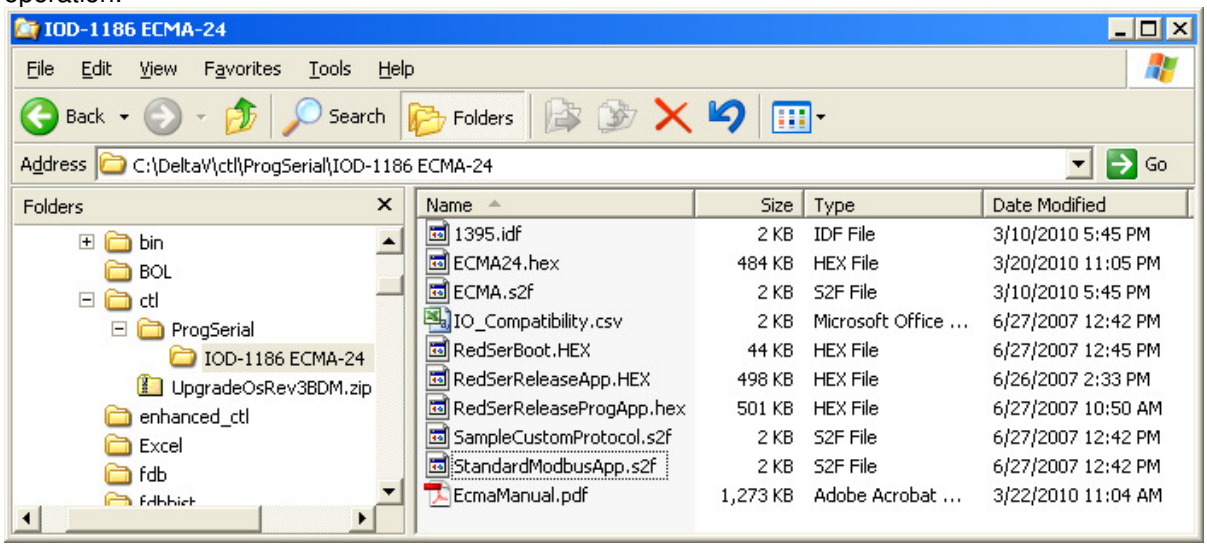


3 Flashing the firmware

The driver software distribution contains 9 files. These files must be copied to the DeltaV directory on your ProPlus Workstation. The path is:

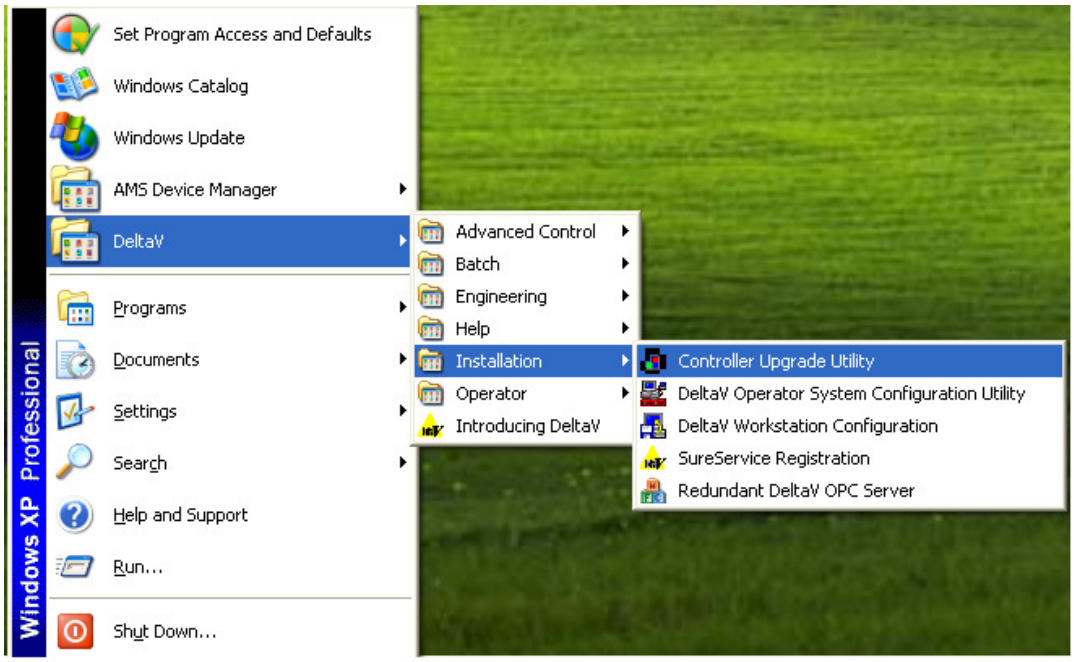
\DeltaV\ctl\ProgSerial\IOD-1186 ECMA-24

Note that you will have to create this subdirectory. The following shows a completed copy operation:

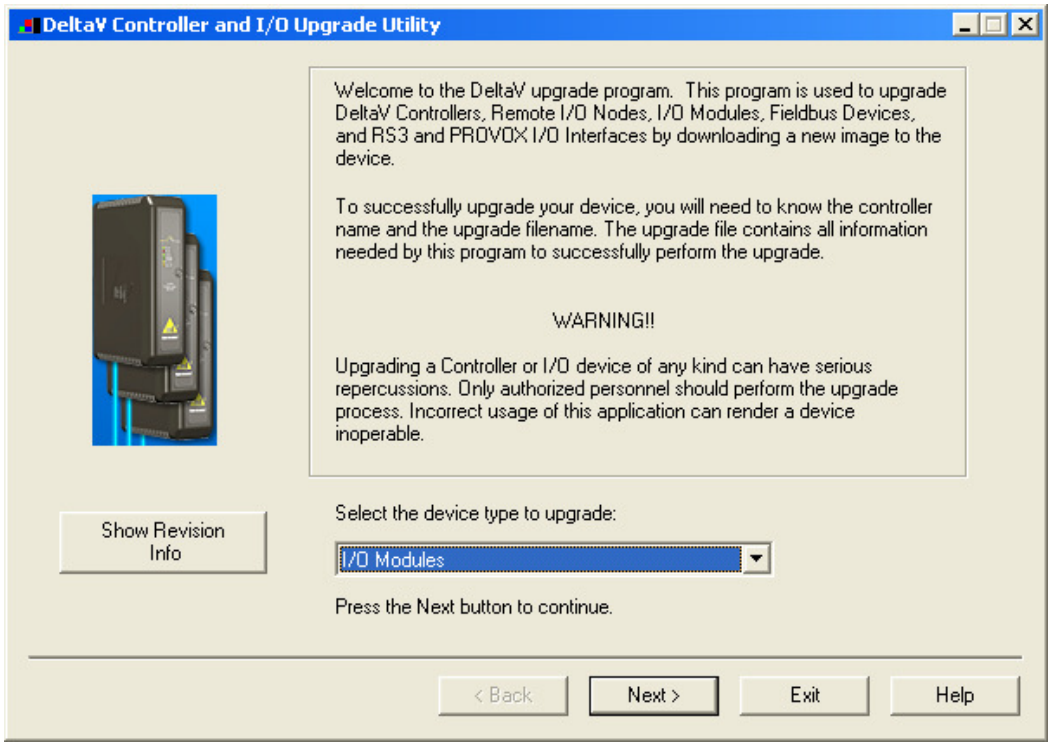


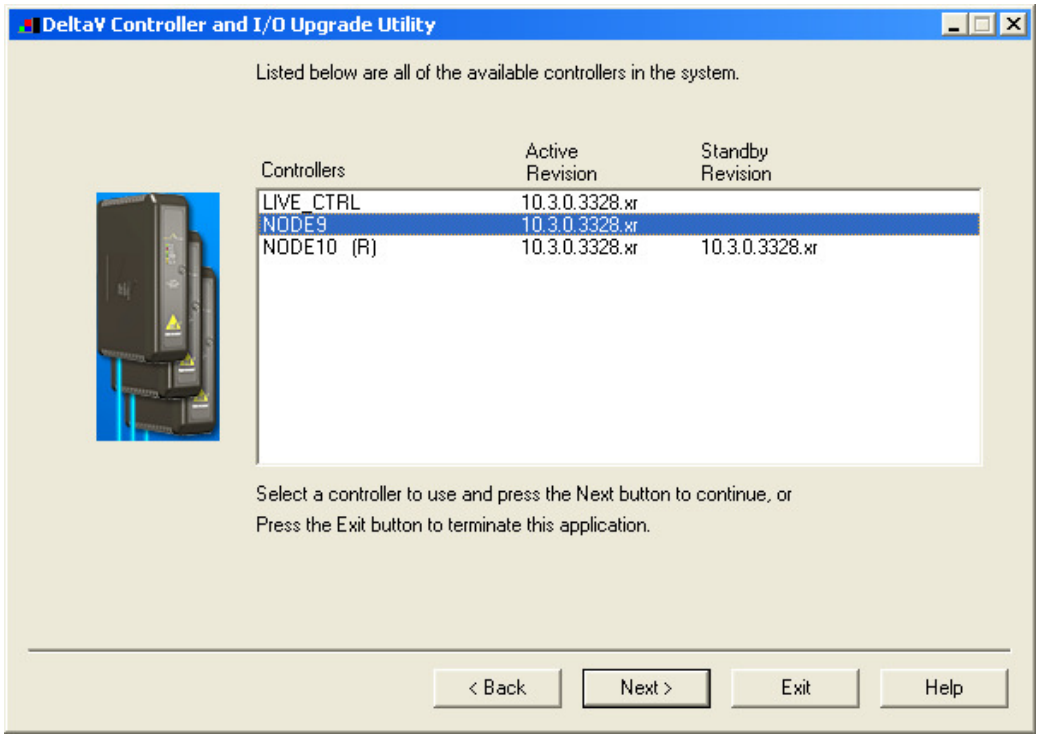
After copy completion, you are ready to program (or upgrade) the Programmable Serial Card with the supplied custom driver software. The steps are as follows:

1. Click on the Start button and select DeltaV-> Installation-> Controller Upgrade Utility as shown below, and the following dialog will appear:



2. Click on the Upgrade I/O Modules radio button, and then click Next.

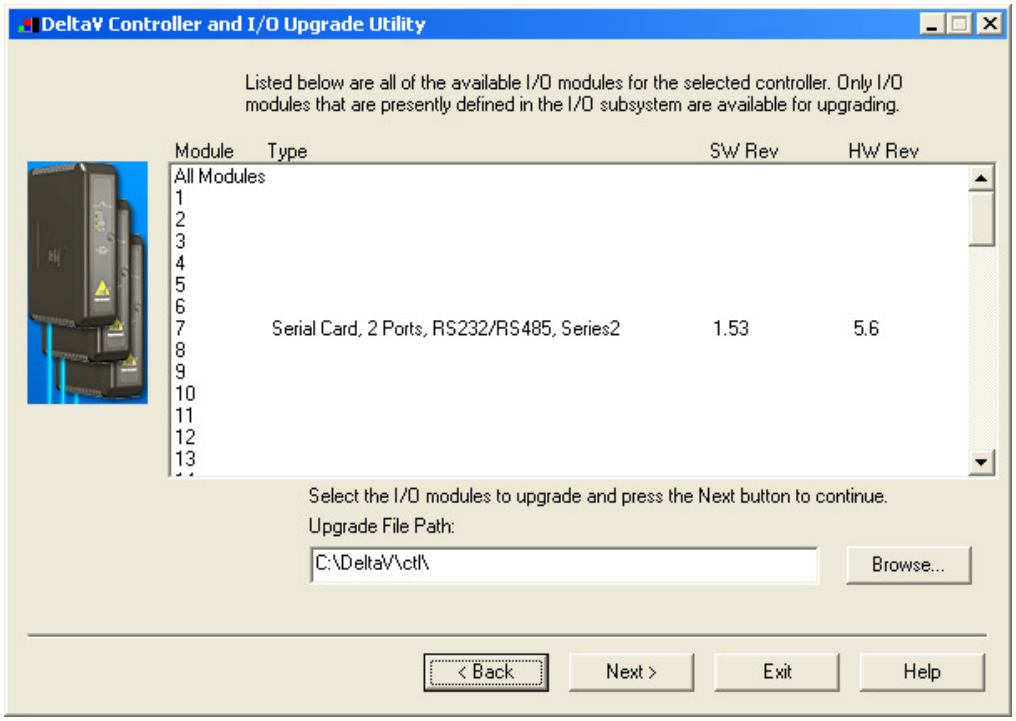




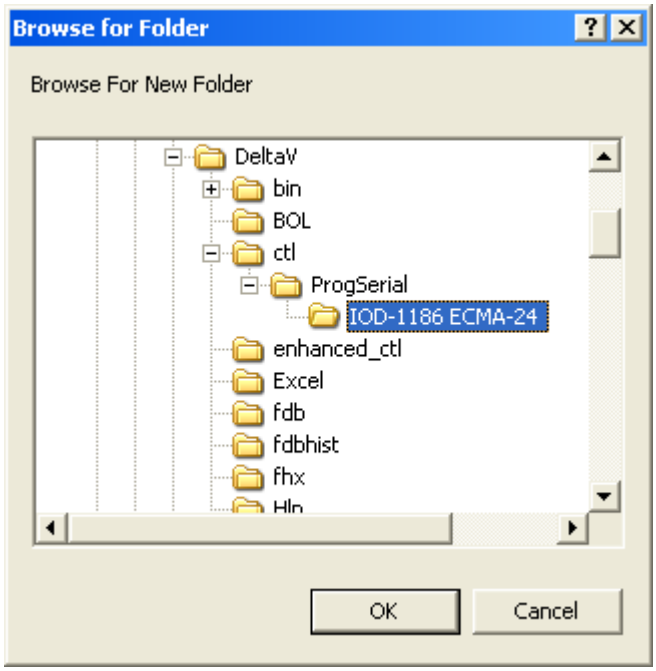
3. The above dialog will appear, listing all the available Controllers in your network. From this dialog, select the appropriate Controller and then Click Next.

4. The following dialog will appear, listing all the I/O modules in your selected Controller. The shown list of I/O modules is an example only. Your list will be different.

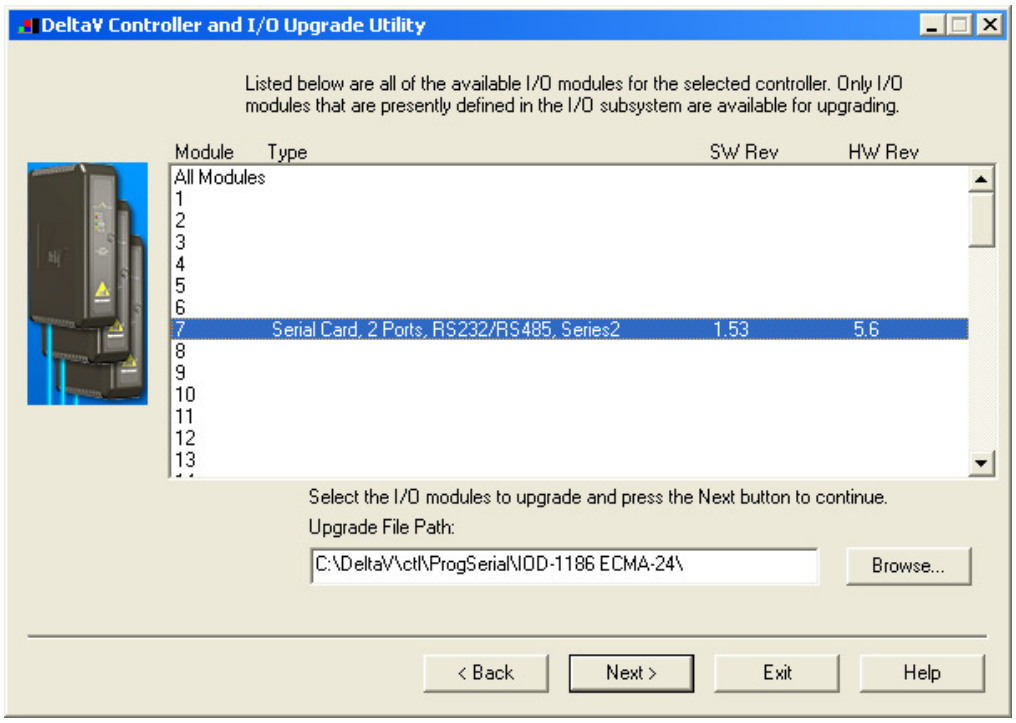
Note: The first time a standard Serial card is upgraded to the ECMA-24 Driver, the dialog will be as shown below. When upgrading an existing Programmable Serial Card, skip Steps 5 and 6, and go to Step 7.



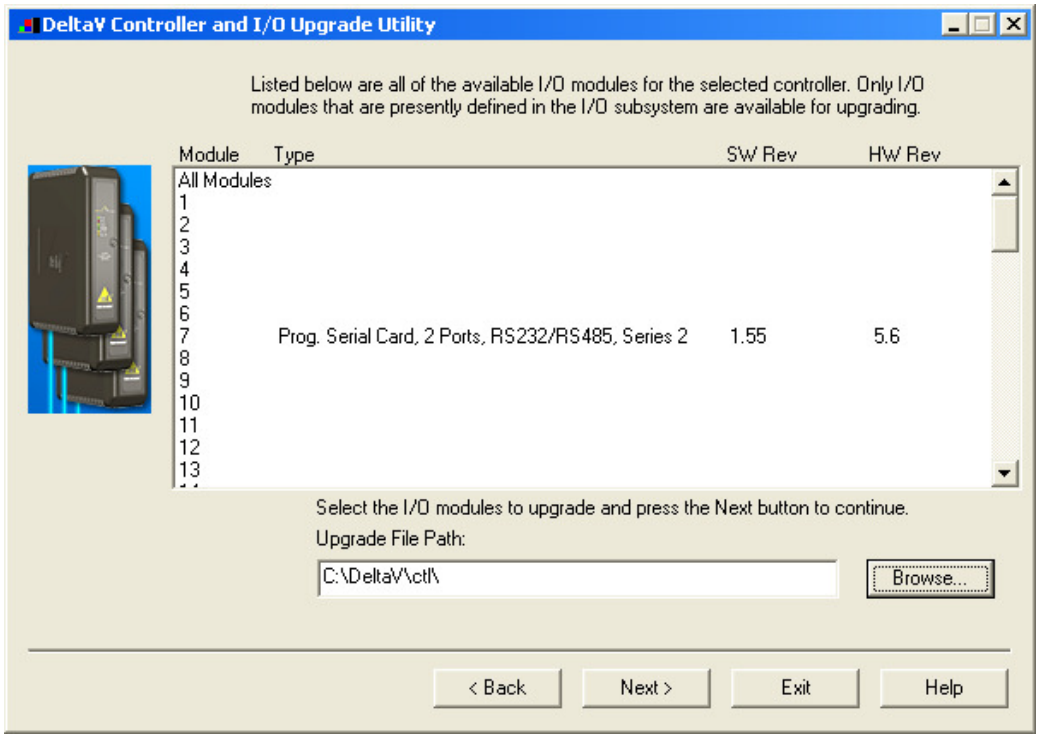
5. Click the Browse button and select the DeltaV path as shown below, and then click Ok. Note that the disk drive could be C or D.



6. Select the I/O module again as shown below and then click Next. Go to Step 9.



7. If you are upgrading an existing Programmable Serial Card, the dialog will be as shown below. From this dialog, select the Programmable Serial Card I/O Module in the list.



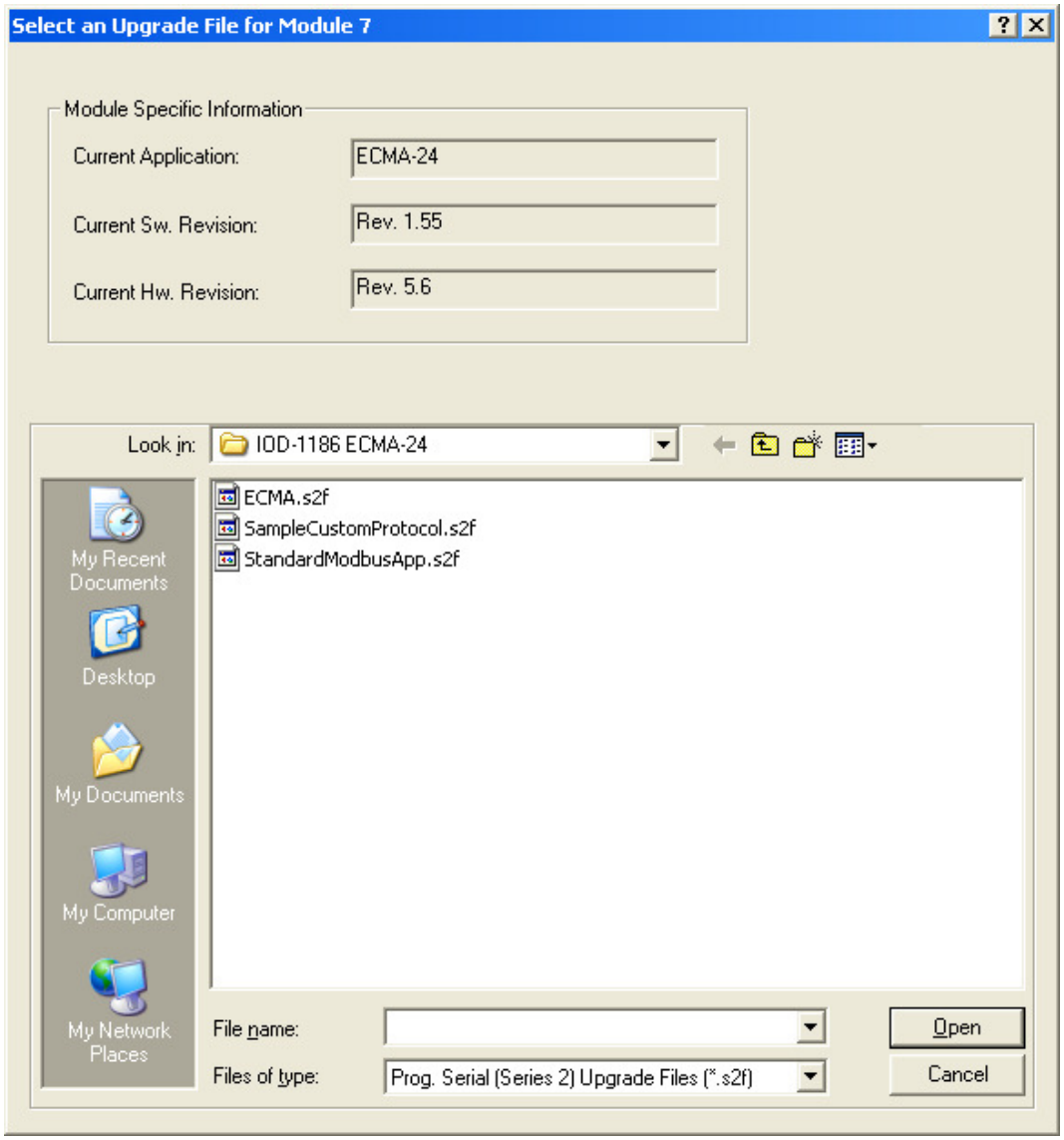
For example, we will select I/O Module 7. This will give you a dialog, from which you will select the file path to where the driver software is located. This path will be:

\DeltaV\ctl\ProgSerial\IOD-1186 ECMA-24

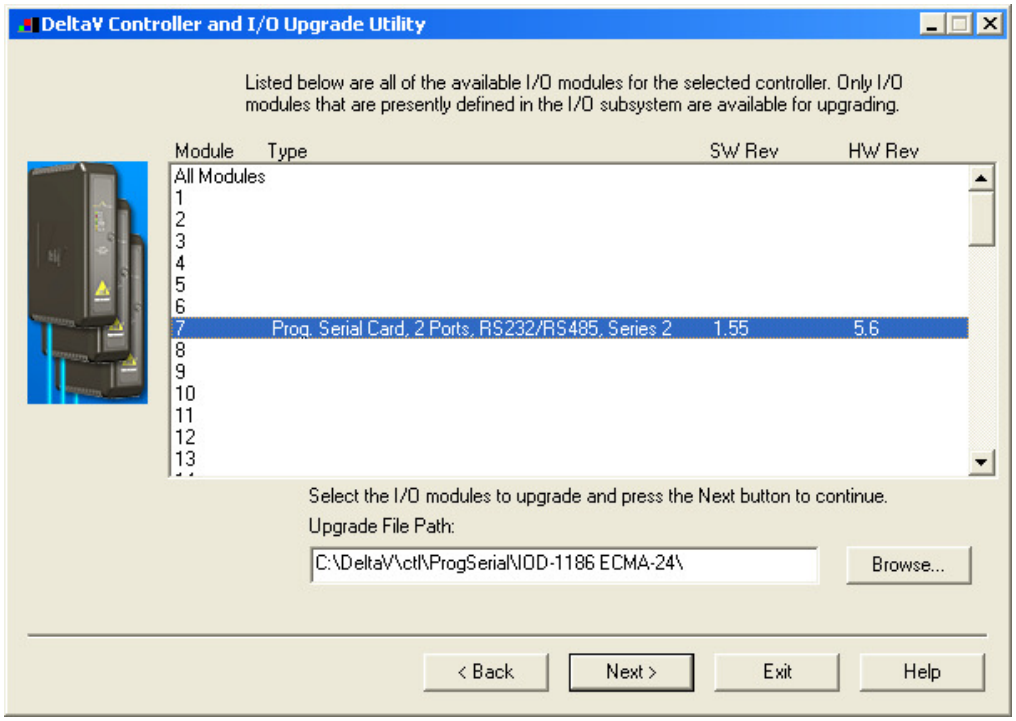
Once you are in the specified directory, you will need to select the following file:

ECMA.S2F

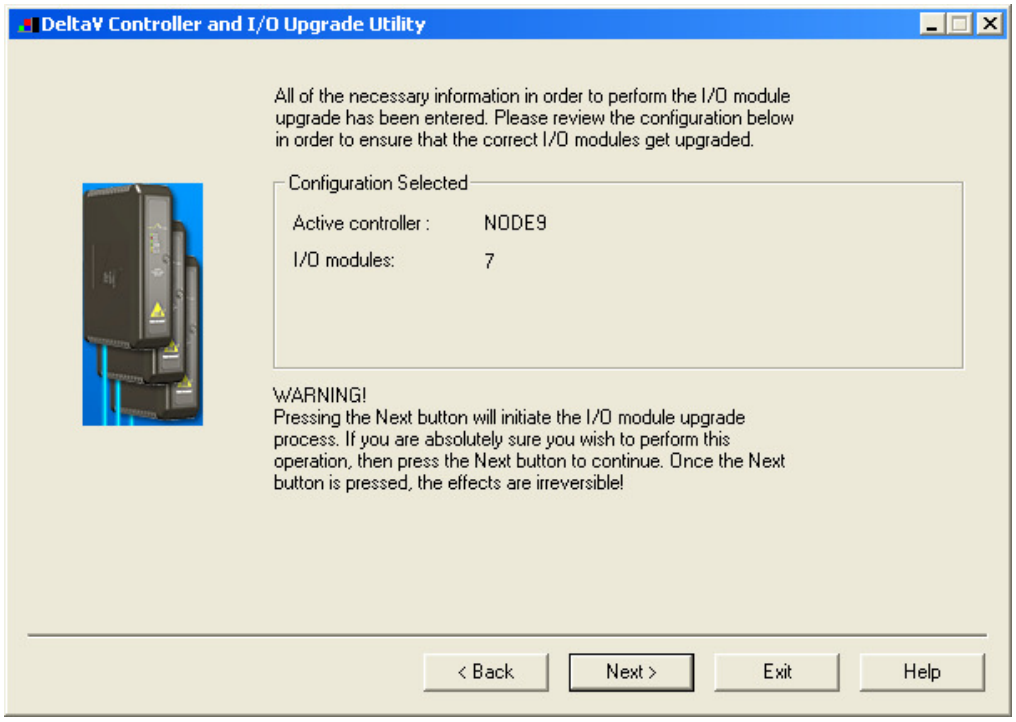
This is shown in the following dialog.



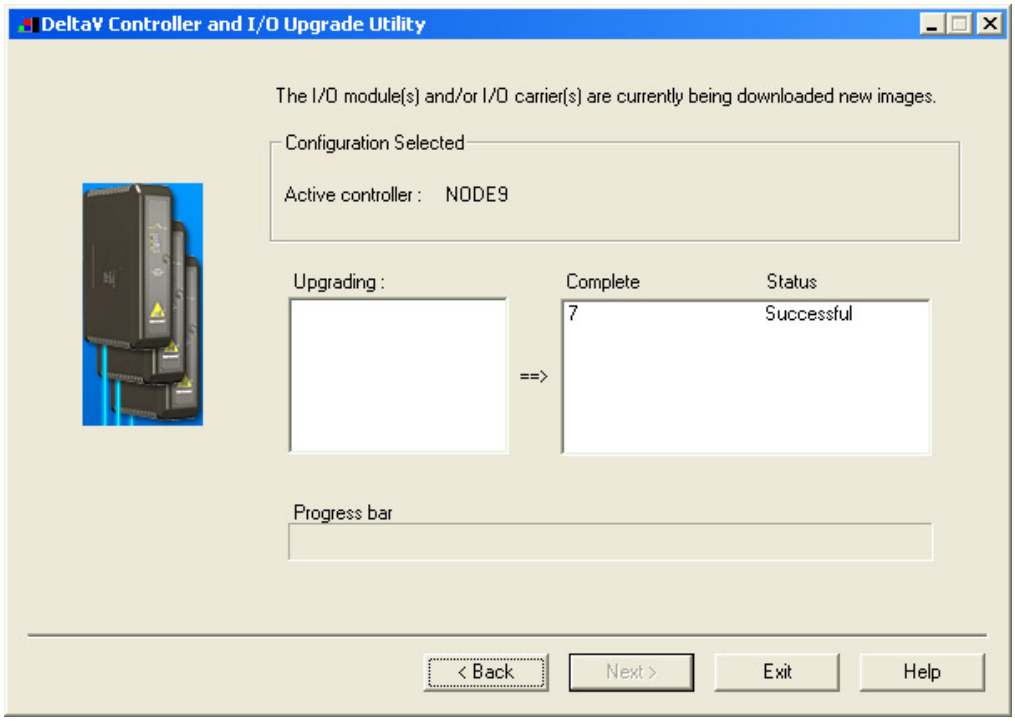
8. After selecting the .S2F file, Click on Open. This dialog will close and you will be back to the following:



9. In this dialog, Click Next again. You will get the following dialog, confirming the Controller and I/O Module to program.



10. Click Next and the I/O Module upgrade process will begin. After completion, you will receive the following dialog, indicating success.

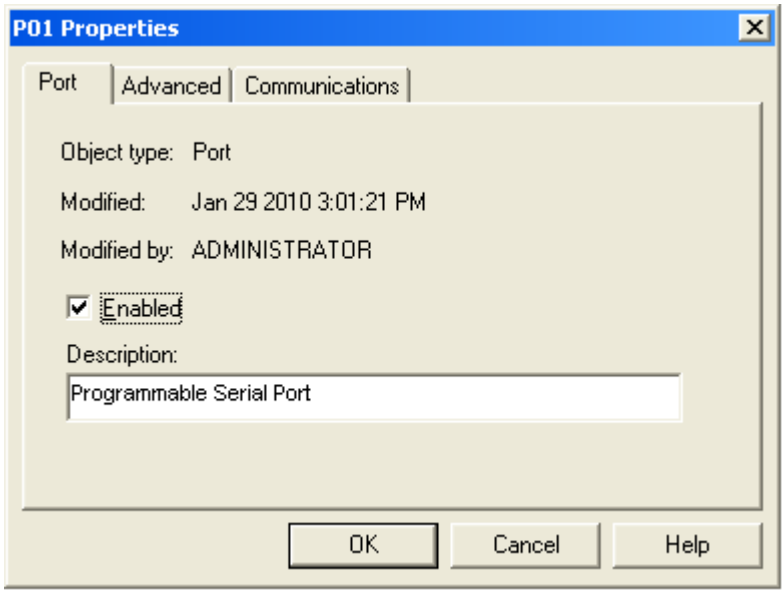


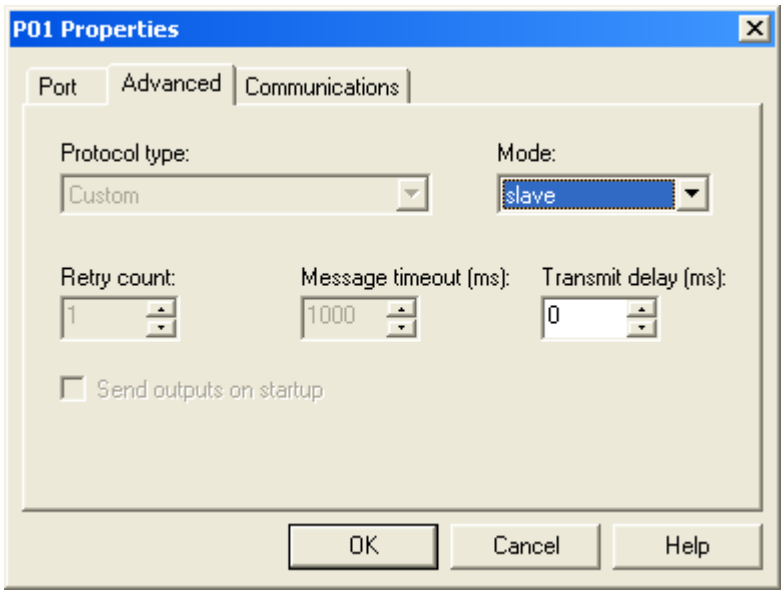
11. This completes the I/O Module upgrade process.

4 CONFIGURATION INFORMATION

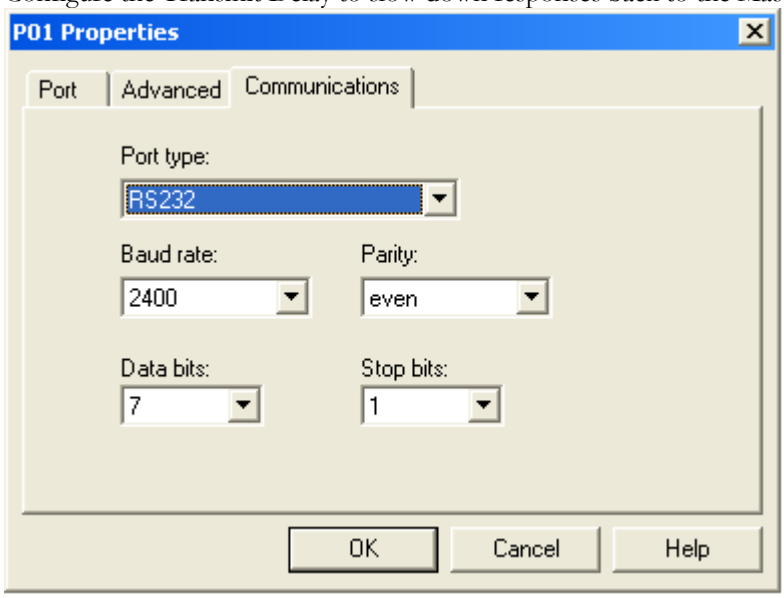
4.1 Port Configuration

First, enable the port. Then click on the Advanced Tab and select Slave. Next, click on the Communications Tab and specify the Port type. The Port type will be RS-232, RS-422/485 Half Duplex (2 wire), or RS-422/485 Full Duplex (4 wire), as required by the external system. Lastly, select the Baud rate, Parity, Data bits and Stop bits parameters; these must match the external device. The following screen shots show the configuration:



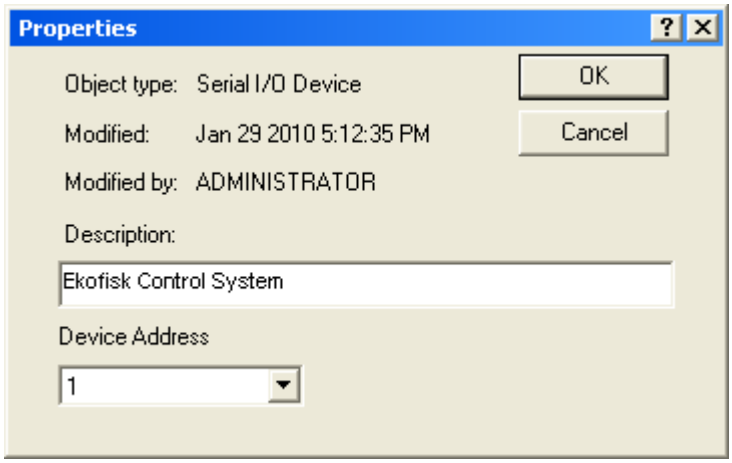


Configure the Transmit Delay to slow down responses back to the Master device.



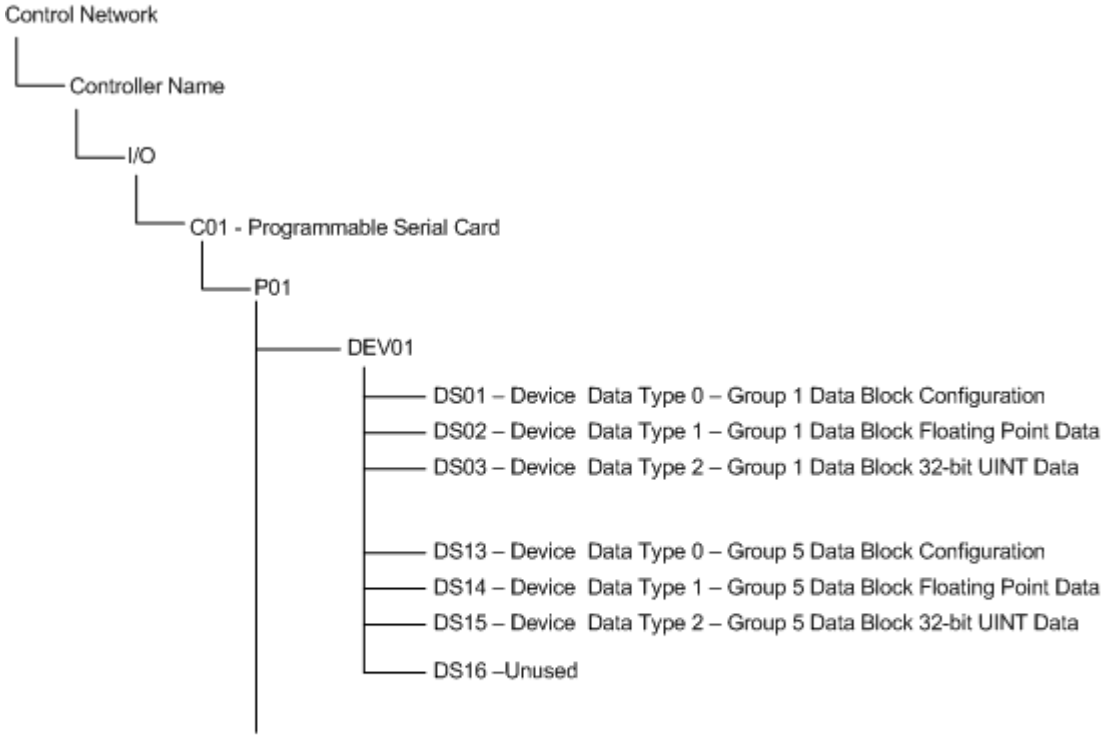
4.2 Device Configuration

Specify a single device and device address. The device address corresponds to the expected Machine ID in the ECMA-24 protocol.



4.3 Dataset Configuration

A fixed dataset architecture is used to specify data block structures and to store dynamic data block information. Datasets are configured in 5 groups of 3 datasets each. The remaining 16th dataset is unused. This provides capacity for 15 data blocks under a single port interface as described below. Dataset usage is as follows:



4.3.1 Device Data Types:

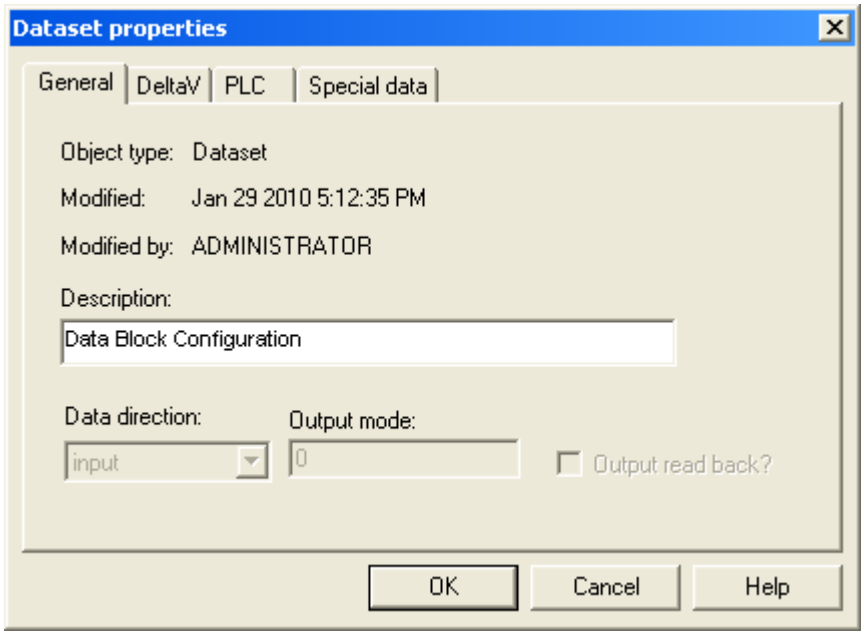
Only the following Device Data Types are valid. Datasets must be configured in order as shown above.

Device Data Type	Description
0	Dataset holds Data Block Configuration. The 3 dataset group comprises Inputs from the external device.
1	Dataset holds Block Floating Point Data
2	Dataset holds Block 32-bit UINT Data
3	Dataset holds Data Block Configuration. The 3 dataset group comprises Outputs to the external device.

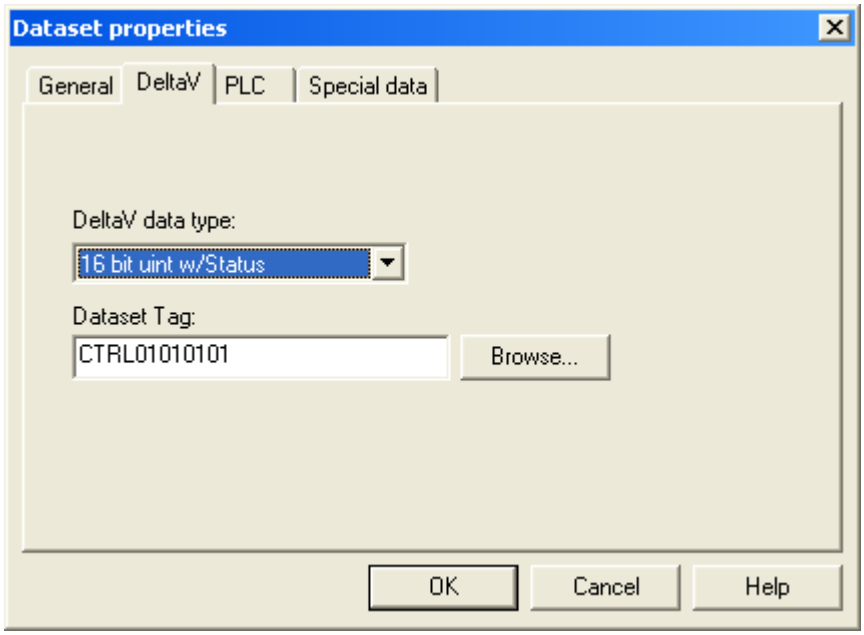
4.3.2 Data Block Configuration:

The Block Configuration dataset is configured as follows. The registers in these datasets are divided into 3 sections of 25 registers each. Each section of 25 registers will correspond to a specific data block to be read or written.

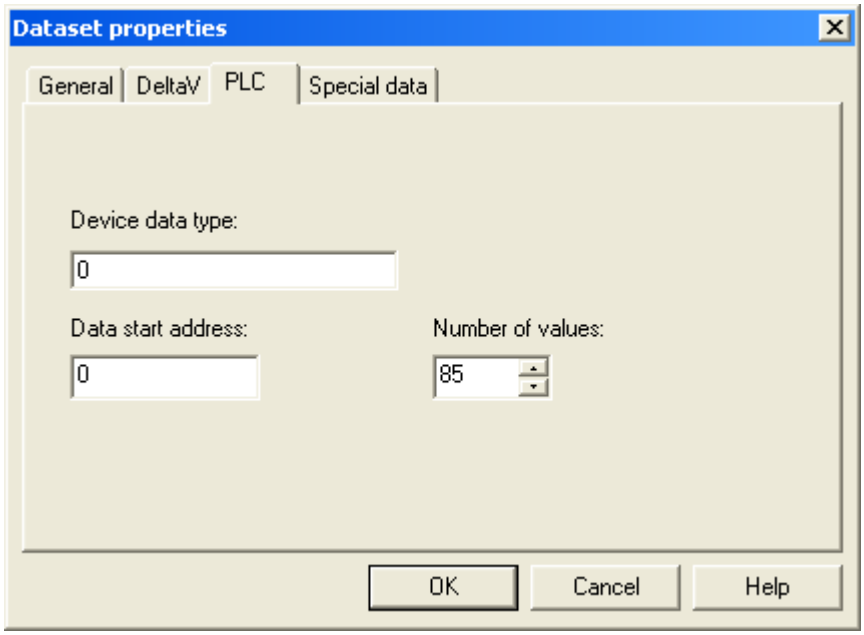
Direction	Input
DeltaV Data Type	16-bit UINT
Device Data Type	0 for read only blocks; 3 for write only blocks.
Start Address	Based on group number. 0, 300, 600, 900, 1200
Number of Values	85
Special Data 1	ID of first data block, Registers 1-25
Special Data 2	ID of second data block, Registers 26-50
Special Data 3	ID of third data block, Registers 51-75
Special Data 4	Unused
Special Data 5	Unused



Because of Slave mode selection at the Port, the Data Direction defaults to Input. Under the DeltaV Tab, the parameters are configured as follows:



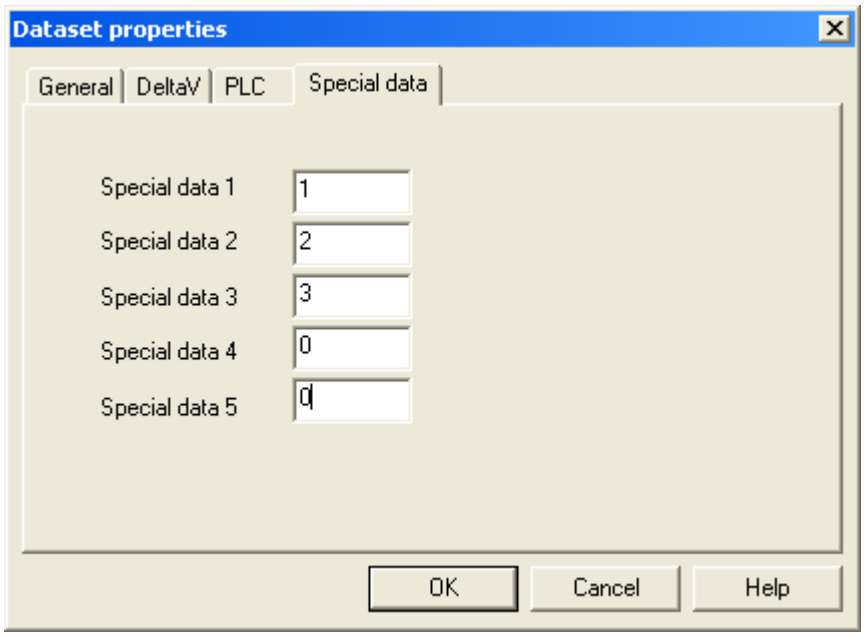
Under the PLC Tab, the parameters shall be as follows:



The Device Data Type value of 0 indicates that all data blocks configured in this group are Input only (data transferred from external device to DeltaV). Configure the Device Data Type as 3 for data blocks which will be written out from DeltaV to the external device.

Registers 1-75 contain the data block configuration values. Registers 80-85 contain status flags for the three data blocks in this group. Each data block has 2 status registers. See Section 4.3.2.2 below.

Under the Special Data Tab shown below, only the Special Data 1, 2 and 3 parameters are used. These values must uniquely identify a data block. That is, these values are used as the Block ID (BID) in communications.



The registers in the Data Block Configuration dataset are sectioned as follows:

Section 1	Section 2	Section 3
R1	R26	R51
R2	R27	R52
R3	R28	R53
R25	R50	R75

Only the first 24 registers of each section are used to define the data block structure. Register values of 0 are ignored. For any other errors, the whole group will be marked as invalid. For example, if the Device Data Type does not match the expected types, or an invalid register number for data storage is given. Maximum number of values which a data block can contain is 24, with any mix of up to 16 integer type and 16 floating point type values.

The 25th register of a section is used to trigger a write to the external device. This register is checked by the driver only if the Device Data Type is 3 (Write Blocks). A user Control Module configured in DeltaV is used to change a Data Blocks dynamic values. The user then writes a 1 into the 25th register of a section to trigger a write. The procedure is as follows:

1. Write new data values to the 32-bit and floating point datasets for a specific data block.

2. Write a 1 to the Write Command register of that data block (R25, R50, or R75).

3. The driver will format a message packet based on the data block configuration.

4. The driver will wait for an invitation to send data. On receipt of invitation, the formatted packet will be sent and the Write command register will be reset to 0.

4.3.2.1 Register Structure in Block Configuration Dataset

The 16-bit register values are specified in low byte, high byte format as follows:

Low Byte	Data Type of value in Data Block. Data Types are enumerated below.
High Byte	Register number where this value will be stored in the 32-bit or Floating Point datasets.

The value configured into the dataset register is (256 * High Byte) + Low Byte.

Preconfigured Data Types, corresponding to expected data bytes in communications packets, and targeted storage dataset type are as follows:

Type	Enumerated Value	Number of expected bytes	Storage
I1	1	1	Into 32-bit dataset
I2	2	2	Into 32-bit dataset
3I2	3	6	Into 32-bit dataset
4I2	4	8	Into 32-bit dataset
I6	5	6	Into 32-bit dataset
I8	6	8	Into 32-bit dataset
I9	7	9	Into 32-bit dataset
F6.1	20	6	Into Floating Point dataset
F6.2	21	6	Into Floating Point dataset
F6.3	22	6	Into Floating Point dataset
F7.1	23	7	Into Floating Point dataset
F7.2	24	7	Into Floating Point dataset
F7.3	25	7	Into Floating Point dataset
F8.1	26	8	Into Floating Point dataset
F8.2	27	8	Into Floating Point dataset
F8.3	28	8	Into Floating Point dataset

The storage register number is a value in the range 1-16. This value is not expected to be duplicated within the same data block, except when referring to the 32-bit vs. Floating point datasets. If duplicated, the second occurrence will overwrite the first.

All data received from the external device is expected to be right justified in the data field, with leading spaces as required.

All data sent to the external system will be right justified in the data field, with leading spaces as required. For Floating Point values, the precision of the fraction, 1, 2 or 3 digits, will be as specified by the selected Data Type.

For example, consider the following two registers:

Value	**Value Parts**	**Description**
261	Low Byte=5; High Byte=1	Expected value is I8, stored in register 1 of 32-bit dataset
276	Low Byte=20; High Byte=1	Expected value is F6.1, stored in register 1 of floating point dataset

4.3.2.2 Data Value Errors

Data sent or received which exceeds its specified format length is transmitted as asterisks. For example, a value of 1234.56 with a format of F6.2 is transmitted as ***.**.

When the driver receives such data, the last known good value received and reported to DeltaV is retained in the dataset register. An error flag bit is asserted and reported to DeltaV as described below.

Since there are three data blocks per group, and each data block potentially contains more than 16 values (maximum allowed is 24), two special status registers are designated per block (6 registers per group) to contain the value status bits. These registers are in the Data Block Configuration dataset.

Data Block	Status Register
1	80 and 81
2	82 and 83
3	84 and 85

Value status bits are represented as a mask. For example, if the data block contains 10 values, then the first 10 bits of the first status register will be used to indicate if any of the received values are in error. A bit value of 0 means no error. A bit value of 1 means the corresponding received value is in error.

For example, consider register 80 value 40:

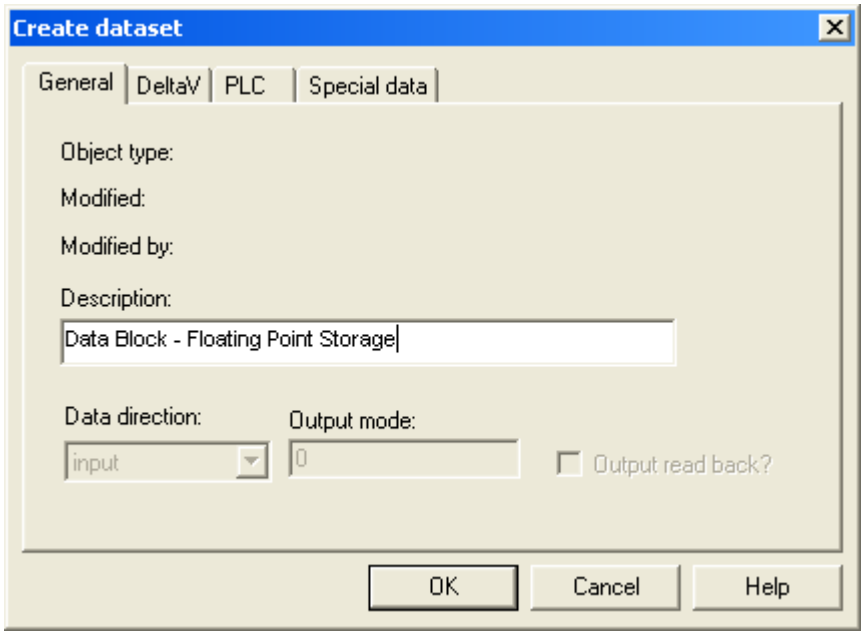
40	Bit 0 = 0	Value 1 is good
	Bit 1 = 0	Value 2 is good
	Bit 2 = 0	Value 3 is good
	Bit 3 = 1	Value 4 is in error
	Bit 4 = 0	Value 5 is good
	Bit 5 = 1	Value 6 is in error
	Bit 6 = 0	Value 7 is good
	Bit 7 = 0	Value 8 is good
	Bit 8 = 0	Value 9 is good
	Bit 9 = 0	Valie 10 is good

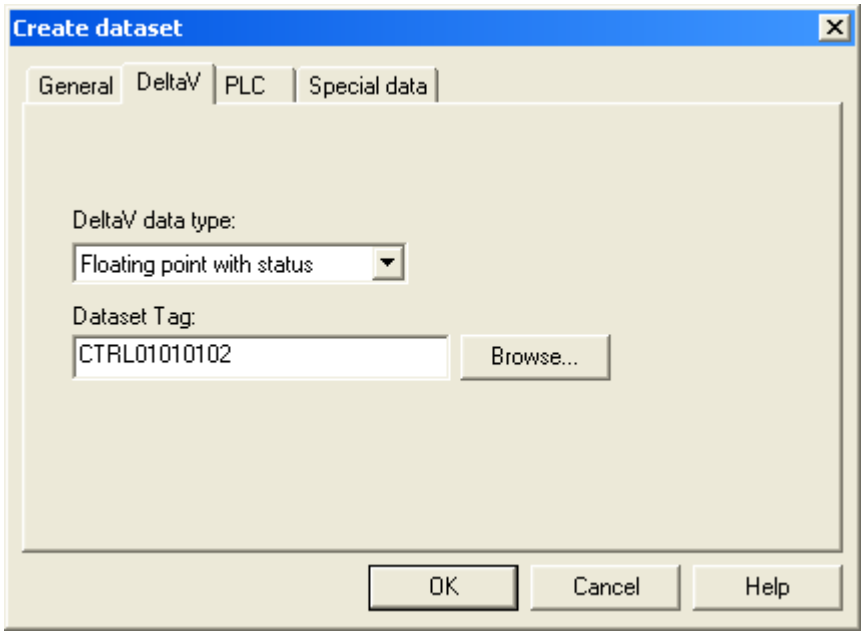
When the data value is not received as ***.**, the driver will report the new good value to DeltaV and clear the corresponding error bit.

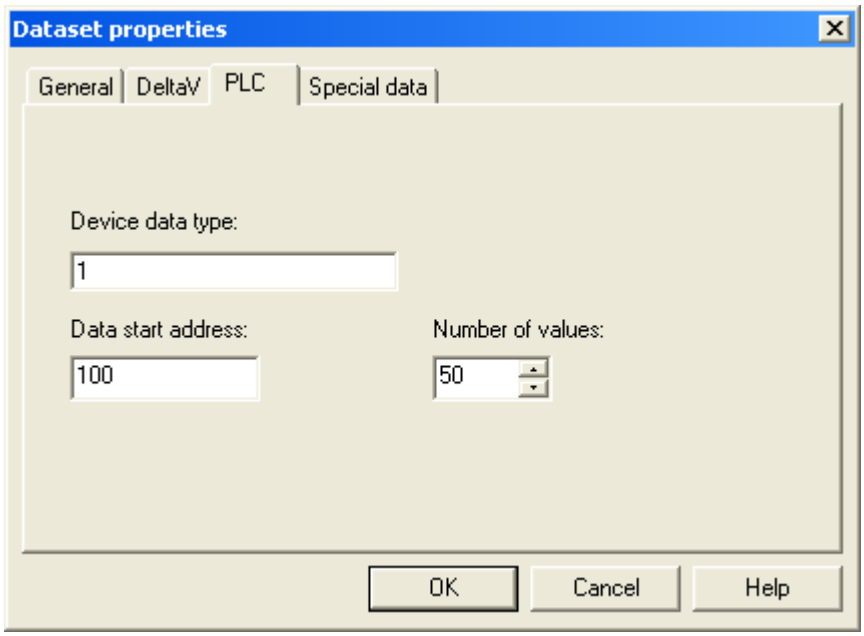
4.3.3 Floating Point Dataset Configuration:

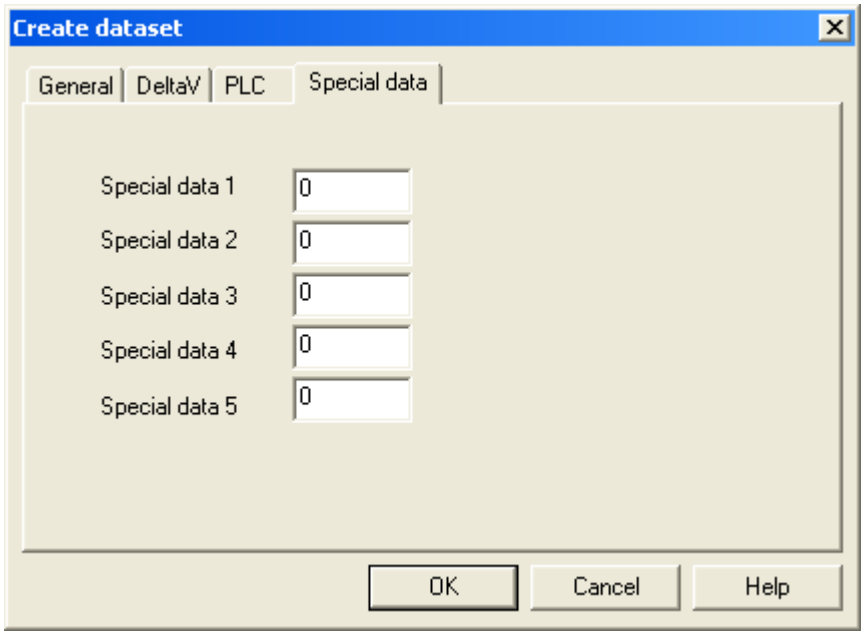
Configure the Floating Point dataset as follows:

Direction	Input
DeltaV Data Type	Floating Point
Device Data Type	1
Start Address	Based on group number. 100, 400, 700, 1000, 1300
Number of Values	50
Special Data 1	Unused
Special Data 2	Unused
Special Data 3	Unused
Special Data 4	Unused
Special Data 5	Unused




Create dataset
General
DeltaV
PLC
Special data
DeltaV data type:
Floating point with status
Dataset Tag:
CTRL01010102
Browse...
OK
Cancel
Help


Dataset properties
General
DeltaV
PLC
Special data
Device data type:
1
Data start address:
100
Number of values:
50
OK
Cancel
Help



The registers in the Floating Point dataset shall be sectioned as follows:

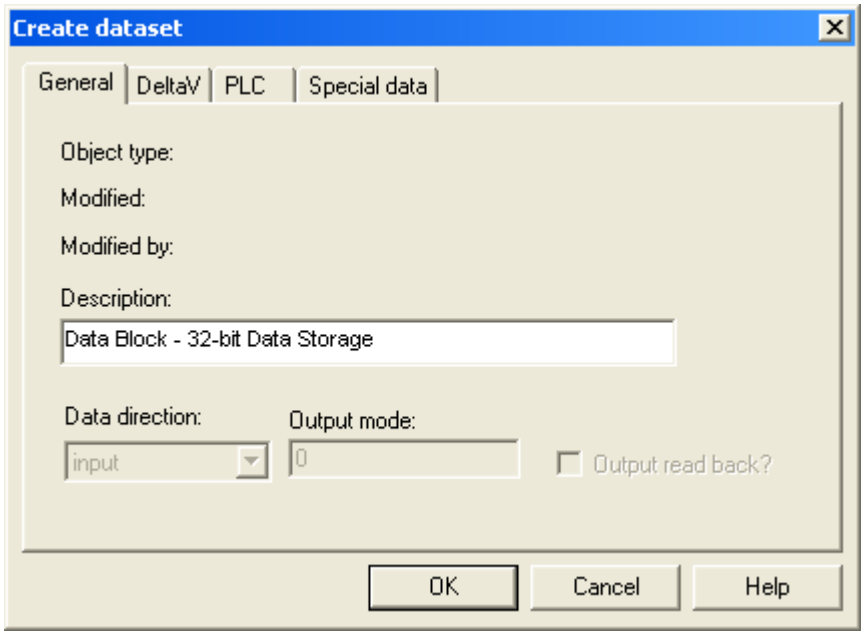
Section 1	**Section 2**	**Section 3**
R1	R17	R33
R2	R18	R34
R3	R19	R35
R16	R32	R48

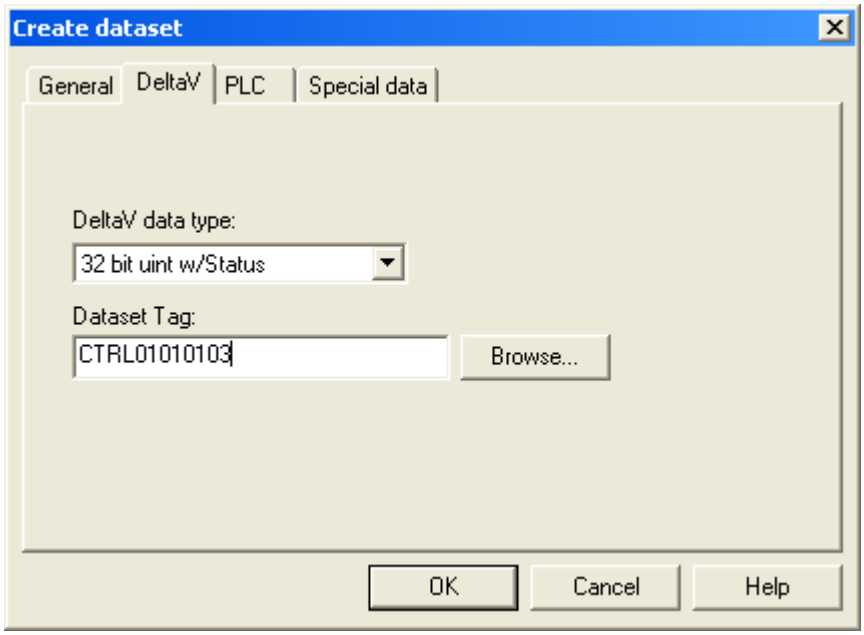
Section 1 belongs to Data Block ID specified in Special Data 1 of the Data Block Configuration dataset. Similarly, Section 2 and 3 belong to Block ID's specified in Special Data 2 and 3.

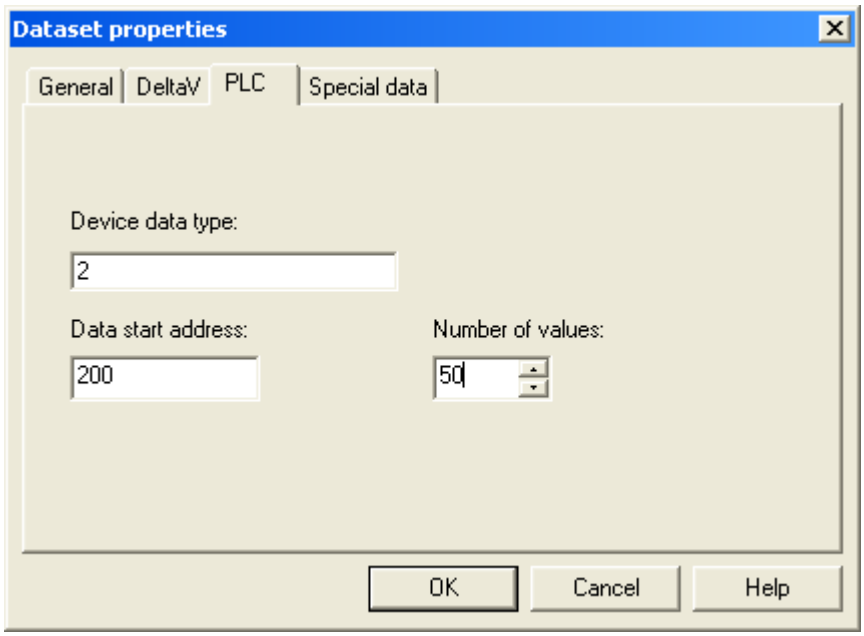
4.3.4 32-Bit UINT Dataset Configuration:

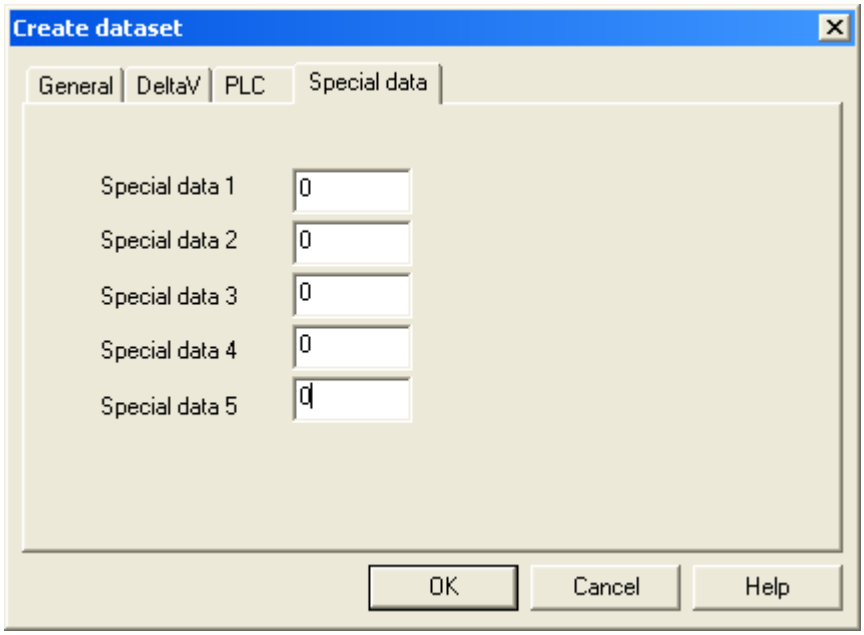
Configure the 32-Bit Integer dataset as follows.

Direction	Input
DeltaV Data Type	32-bit UINT
Device Data Type	2
Start Address	Based on group number. 200, 500, 800, 1100, 1400
Number of Values	50
Special Data 1	Unused
Special Data 2	Unused
Special Data 3	Unused
Special Data 4	Unused
Special Data 5	Unused




Create dataset
General
DeltaV
PLC
Special data
DeltaV data type:
32 bit uint w/Status
Dataset Tag:
CTRL01010103
Browse...
OK
Cancel
Help


Dataset properties
General
DeltaV
PLC
Special data
Device data type:
2
Data start address:
200
Number of values:
50
OK
Cancel
Help



The registers in the 32-bit dataset are sectioned as follows:

Section 1	Section 2	Section 3
R1	R17	R33
R2	R18	R34
R3	R19	R35
R16	R32	R48

Section 1 belongs to Data Block ID specified in Special Data 1 of the Data Block Configuration dataset. Similarly, Section 2 and 3 belong to Block ID's specified in Special Data 2 and 3.

4.3.5 Example - Read Data Block Configuration:

The following is an example of a data block configuration for a data block received from the external device. The Data block ID is 1.

Data Direction	Input
DeltaV Data Type	16-bit UINT
Device Data Type	0
Start Address	0
Number of Values	85
Special Data 1	1 (ID of Data Block)
Special Data 2	0 – Unconfigured
Special Data 3	0 – Unconfigured
Special Data 4	0 – Unconfigured
Special Data 5	0 – Unconfigured

Data Block Configuration Registers will be as follows. Value is calculated as (256 * High Byte) + Low Byte.

Section 1	Low Byte	High Byte	Value	Description
R1	3	1	259	Time stamp – Type 3I2
R2	6	2	518	Current gross volume flowrate – Type I8
R3	6	3	774	Current standard volume flowrate – Type I8
R4	21	1	277	Average Temperature – Type F6.2
R5	23	2	535	Average Pressure – Type F7.1
R6	28	3	796	Average Density – Type F8.3
R7	6	4	1030	Gross Volume total previous hour – Type I8
R8	6	5	1286	Gross Volume total current hour – Type I8
R9	6	6	1542	Mass total previous hour – Type I8
R10	6	7	1798	Mass total current hour – Type I8
R11	6	8	2054	Volume total previous day – Type I8
R12	6	9	2310	Volume total current day – Type I8
R13	6	10	2566	Mass total previous day – Type I8
R14	6	11	2822	Mass total current day – Type I8
R15				Unused
R16				Unused
R17				Unused
R18				Unused
R19				Unused
R20				Unused
R21				Unused
R22				Unused
R23				Unused
R24				Unused

The datasets containing dynamic block data for the above configuration are as follows.

32-bit Integer Dataset		Floating Point Dataset	
R1	Time Stamp	R1	Average Temperature
R2	Current gross volume flowrate	R2	Average Pressure
R3	Current standard volume flowrate	R3	Average Density
R4	Gross volume total previous hour	R4	Unused
R5	Gross volume total current hour	R5	Unused
R6	Mass total previous hour	R6	Unused
R7	Mass total current hour	R7	Unused
R8	Volume total previous day	R8	Unused
R9	Volume total current day	R9	Unused
R10	Mass total previous day	R10	Unused
R11	Mass total current day	R11	Unused
R12	Unused	R12	Unused
R13	Unused	R13	Unused
R14	Unused	R14	Unused
R15	Unused	R15	Unused
R16	Unused	R16	Unused

4.3.6 Example - Write Data Block Configuration:

The following is an example of a data block configuration for a data block sent from DeltaV to the external device. The Data block ID is 1.

Data Direction	**Input**
DeltaV Data Type	16-bit UINT
Device Data Type	3
Start Address	0
Number of Values	85
Special Data 1	1 (ID of Data Block)
Special Data 2	0 - Unconfigured
Special Data 3	0 – Unconfigured
Special Data 4	0 – Unconfigured
Special Data 5	0 – Unconfigured

Data Block Configuration Registers will be as follows. Value is calculated as (256 * High Byte) + Low Byte.

Section 1	Low Byte	High Byte	Value	Description
R1	20	1	276	Oil pipeline inlet pressure – Type F6.1
R2	21	2	533	Oil pipeline inlet temperature – Type F6.2
R3	5	1	261	Teesside storage average level – I6
R4	1	2	513	Oil pipeline inlet ESD valve open – Type I1
R5	1	3	769	Oil pipeline inlet ESD valve closed – Type I1
R6	1	4	1025	Sphere 1 high level alarm – Type I1
R7	1	5	1281	Sphere 2 high level alarm – Type I1
R8	1	6	1537	Sphere 3 high level alarm – Type I1
R9	1	7	1793	Sphere 4 high level alarm – Type I1
R10	1	8	2049	Sphere high level shutdown initiated – Type I1
R11	1	9	2305	Pipeline manual shutdown initiated – Type I1
R12	1	10	2561	Pipeline onshore valve closure initiated – Type I1
R13	5	11	2821	Sphere 1 storage level – Type I6
R14	5	12	3077	Sphere 2 storage level – Type I6
R15	5	13	3333	Sphere 3 storage level – Type I6
R16	5	14	3589	Sphere 4 storage level – Type I6
R17				Unused
R18				Unused
R19				Unused
R20				Unused
R21				Unused
R22				Unused
R23				Unused
R24				Unused

The datasets containing dynamic block data for the above configuration is as follows.

32-bit Integer Dataset		Floating Point Dataset	
R1	Teesside storage average level	R1	Oil pipeline inlet pressure
R2	Oil pipeline inlet ESD valve open	R2	Oil pipeline inlet temperature
R3	Oil pipeline inlet ESD valve closed	R3	Unused
R4	Sphere 1 high level alarm	R4	Unused
R5	Sphere 2 high level alarm	R5	Unused
R6	Sphere 3 high level alarm	R6	Unused
R7	Sphere 4 high level alarm	R7	Unused
R8	Sphere high level shutdown initiated	R8	Unused
R9	Pipeline manual shutdown initiated	R9	Unused
R10	Pipeline onshore valve closure initiated	R10	Unused
R11	Sphere 1 storage level	R11	Unused
R12	Sphere 2 storage level	R12	Unused
R13	Sphere 3 storage level	R13	Unused
R14	Sphere 4 storage level	R14	Unused
R15	Unused	R15	Unused
R16	Unused	R16	Unused

5 Operational Check

5.1 Scope

The following sections provide some assistance to ensure the interface is working properly.

5.2 Verify Hardware and Software Version Number

The user can verify that the Special ECMA-24 driver has been installed using the DeltaV Diagnostics tool. The Diagnostics tool will show the Hardware Revision No. (HwRev) and the Software Revision No. (SwRev).

To begin the DeltaV Diagnostic tool select Start-> DeltaV-> Operator-> Diagnostics. In the Diagnostics tool expand the Controller, I/O and then double click on the Programmable Serial Interface Card that has the driver installed.

The following information will be displayed:

:	:	:
HwRev	Hardware Revision	1.1 (or later)
SwRev	Software Revision	P1.55 (or later)

5.3 Verify Configuration

- Verify port configuration: The serial port must be enabled. User needs to make sure communication settings such as baud rate, parity, and number of data bits match the field device settings.

- Verify dataset configuration: The datasets configured must be as shown above.

5.4 Verify I/O Communication With Control Studio

User can create I/O modules in the control studio to verify correct values are read from the PSIC. For AI and DI data, the values should be changed in the field device and verified that the new data are correctly reported in DeltaV. Similarly, verify that the AO and DO data is being written correctly from DeltaV to the field device.

5.5 Using Diagnostics

- Verify PSIC communication: Select the PSIC on Diagnostics and press the right mouse button. Select Display Real -Time Statistics from the drop down menu. If the Programmable Serial Interface Card is functioning then the user will see the Valid Responses counter and the Async and/or Sync Transactions counters incrementing. There will not be any error counting up.

- Verify port statistics: Select the Port on the Programmable Serial Interface Card and press the right mouse button. Then select Display Port Statistics form the drop down menu. Verify that the port communications statistics are being displayed properly and are counting as expected for the protocol's functionality.

- Verify dataset values: Select a dataset and press the right mouse button. Select View Dataset Registers from the Drop down window. Verify that the dataset values are displayed as expected.

- Verify that there are no errors at the dataset level. The driver asserts the following errors. Note that if any error is found, the driver suspends its functions until a completely error free download is received from the Controller.

Error Message	Description
Invalid Redundant configuration	This driver works only in Simplex mode.
Invalid Configuration Dataset	The Block Configuration dataset is expected to be configured as 16-bit UINT. This applies to datasets of Device Data Type 0 and 3.
Invalid Floating Point Dataset	The second dataset in a group is expected to be configured as Floating Point.
Invalid 32-Bit Integer Dataset	The third dataset in a group is expected to be configured as 32-bit UINT.
Invalid data group configured	The data group is invalid or partially configured.
Invalid Master Mode configured	The driver functions only in Slave mode.
Invalid block ID > 99	Block ID's are expected to be in the range 1-99.
Invalid Blk X, Reg Y Value Z	This error is asserted on the configuration dataset, and is generated if an invalid data block configuration value is encountered. The X is the specific Block ID in error, and the Y and Z are the specific register number and invalid value in the dataset. Correct the value and download the card.

5.6 LED Indication

The Yellow LED for the port should be on solid when all communications on that port are valid. The Yellow LED should be blinking if there is some valid communications and some communications with errors on that port. The Yellow LED should be OFF if there are no valid communications on that port.

6 DeltaV–Field Device Electrical Interface

The electrical interface between DeltaV and field devices conforms to the RS-232 and RS-422/485 standards.

Each PSIC has 2 ports, which function independently. The distance between the serial card and the field device can be as much as 5000 feet, per the RS-422/485 standard. When using RS-232, the distance is limited to 50 feet. Section 6.1 shows the pin assignments for the PSIC serial terminal block.

6.1 Pin Assignments for DeltaV PSIC

RS-232 Standard

Terminal Number	Signal Description
1	Port 1 - Isolated Ground (GND)
2	Unused
3	Port 1 – Transmit Data (TxD)
4	Unused
5	Port 1 – Receive Data (RxD)
6	Unused
7	Port 1 – Data Terminal Ready (DTR)
8	Port 1 – Data Set Ready (DSR)
9	Port 2 - Isolated Ground (GND)
10	Unused
11	Port 2 – Transmit Data (TxD)
12	Unused
13	Port 2 – Receive Data (RxD)
14	Unused
15	Port 2 – Data Terminal Ready (DTR)
16	Port 1 – Data Set Ready (DSR)

RS-422/485 Half Duplex Standard

Terminal Number	Signal Description
1	Port 1 - Isolated Ground (GND)
2	Port 1 - Data +
3	Unused
4	Port 1 - Data -
5	Unused
6	Unused
7	Unused
8	Unused
9	Port 2 - Isolated Ground (GND)
10	Port 2 – Data +
11	Unused
12	Port 2 - Data -
13	Unused
14	Unused
15	Unused
16	Unused

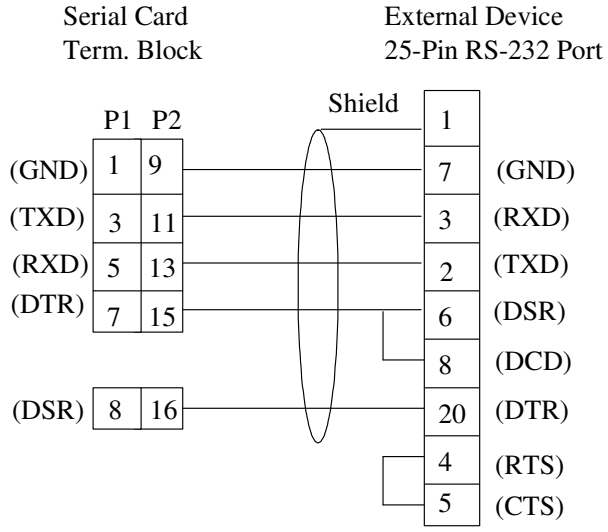
RS-422/485 Full Duplex Standard

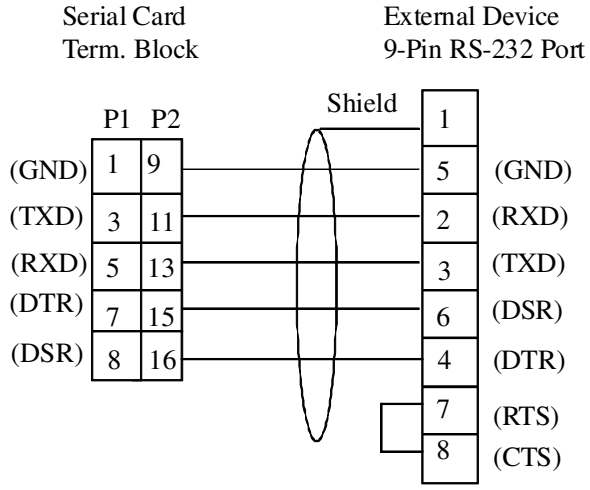
Terminal Number	Signal Description
1	Port 1 - Isolated Ground (GND)
2	Port 1 – TxD +
3	Unused
4	Port 1 – TxD -
5	Unused
6	Port 1 – RxD +
7	Unused
8	Port 1 – RxD -
9	Port 2 - Isolated Ground (GND)
10	Port 2 – TxD +
11	Unused
12	Port 2 – TxD -
13	Unused
14	Port 2 – RxD +
15	Unused
16	Port 2 – RxD -

6.2 Wiring Connections

In general, the figure below shows the connections between the external device and the PSIC termination block. In some cases, RxD and TxD signals need to be swapped to create a NULL cable. This can be done easily at the PSIC termination block.

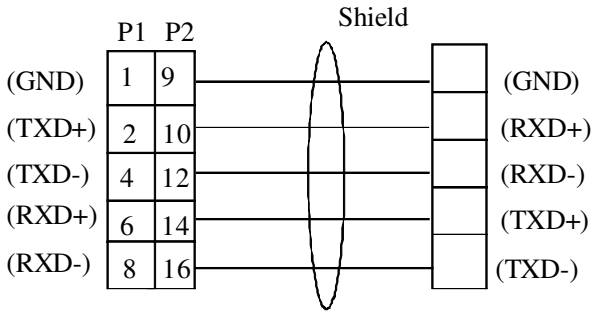
NOTE: If the device uses RS-232 and does not supply DTR and DSR, insert jumpers on the serial card term block between screw term 7 and 8, and 15 and 16. This eliminates the need for wire connections between these screw terms and the field device.





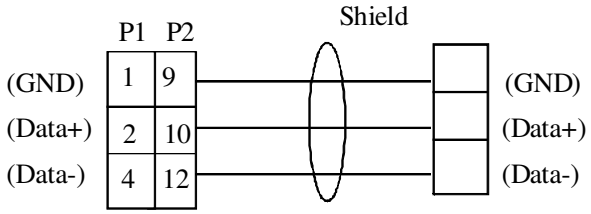
Serial Card
Term. Block

External Device
RS-422/485 Full Duplex



Serial Card
Term. Block

External Device
RS-422/485 Half Duplex



7 Technical Support

For technical support or to report a defect, please give MYNAH Technologies a call at (636) 681-1555. If a defect is discovered, please document it in as much detail as possible and then fax your report to us at (636) 681-1660.

You can also send us your questions via e-mail. Our addresses are:

support@mynah.com

Thank you for using DeltaV.