



DSX Soft I/O Configuration

Overview

The DSX Soft I/O product will allow you to integrate different systems together within the WinDSX software. Now you can use one software package to monitor and control multiple building systems. You can even program action/reaction linking events between diverse systems.

Operation

The DSX Soft I/O program provides a virtual list of inputs and outputs that can be used to accept information IN to the DSX panel network from other systems or send information OUT of the DSX network to other systems.

The DSX Soft I/O program typically requires a dedicated PC that takes the place of controllers on the DSX RS-485 panel network. Basically the Soft I/O program communicates to the Master controller just like Slave controllers would. The Soft I/O program can represent multiple control panels simultaneously. Each control panel (device) that the Soft I/O program represents is capable of 8 soft outputs and 16 soft inputs.

Within the Soft I/O program you define soft inputs and soft outputs. Soft inputs are defined to accept data from other systems and then reflect that data into the DSX control panel network. Soft outputs are defined to transmit data to other systems in response to commands from the DSX control panel network.

The systems Soft IO should communicate with must have the ability to transmit and/or receive ASCII command strings to exchange data.

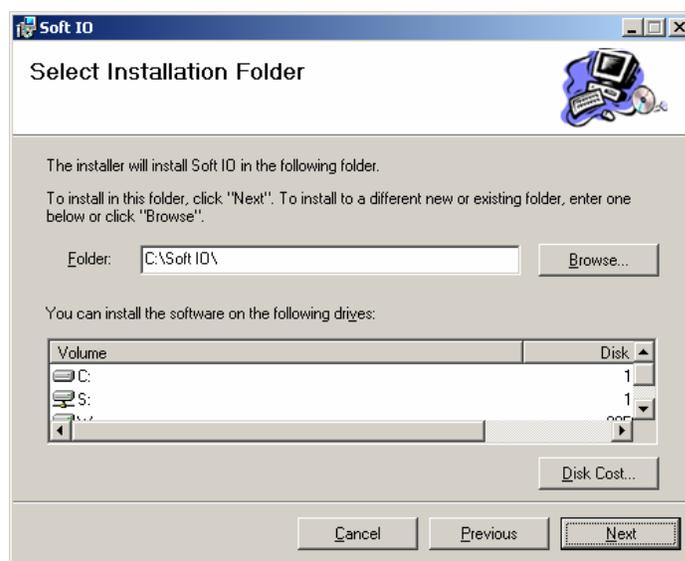
Installing Soft IO

The Soft IO computer must meet the minimum specifications described in the WinDSX PC Requirements document. It is necessary to purchase and register each copy of Soft IO to be used.

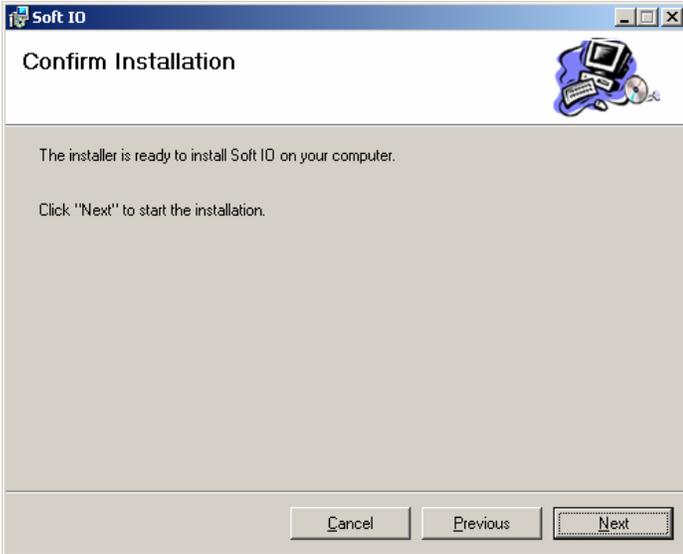
1. To install the Soft I/O program navigate to the Soft IO.msi file and double click.



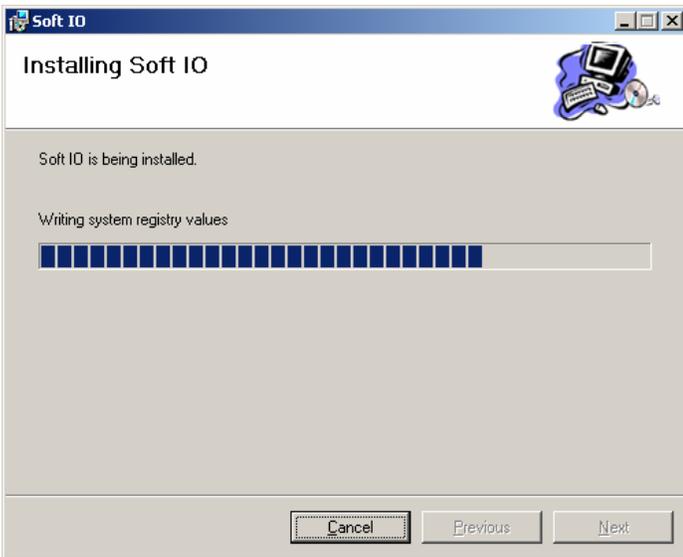
2. Select the existing folder for Soft IO to be installed to or use the default of C:\Soft IO\ and then click on Next.



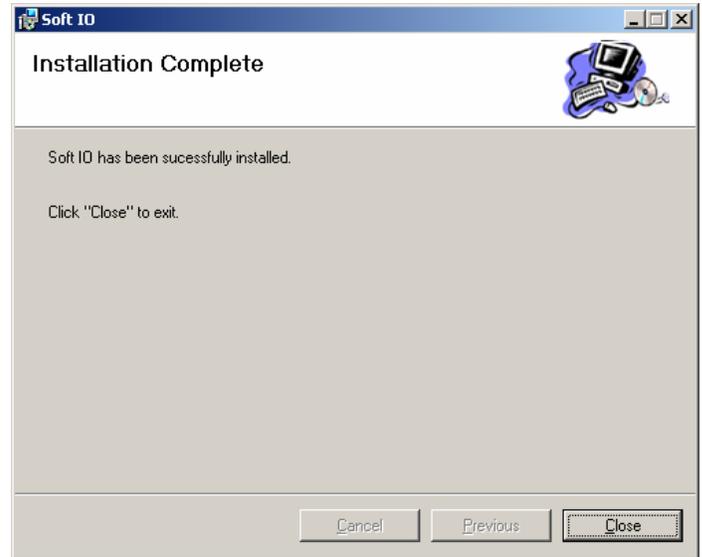
- Click Next on the Installation Confirmation screen.



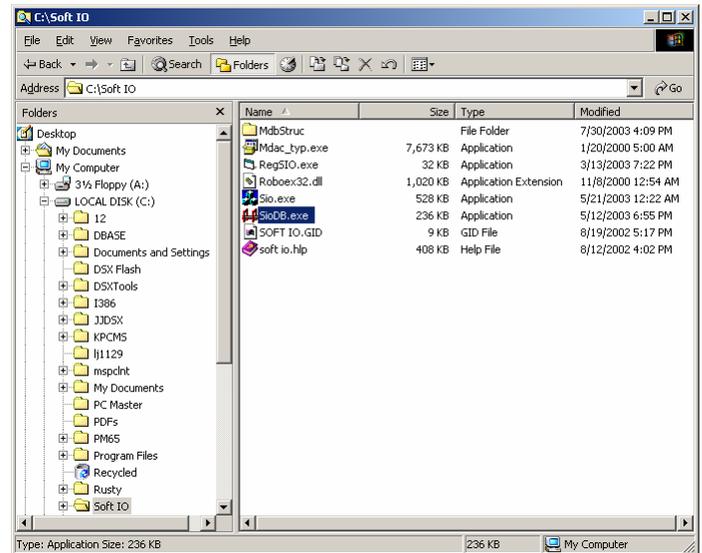
- The installation progression screen is shown next.



- Once the install is complete click on Close.



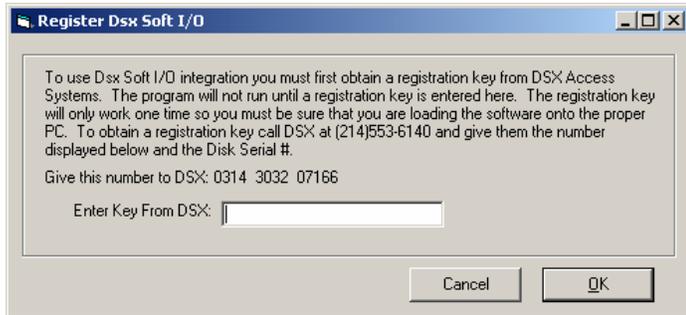
- Navigate to the Soft IO folder and double click on the SiODB.exe file. The first time you run SiODB you will be prompted to register the software. This must be done before configuration can begin.



Note!!!

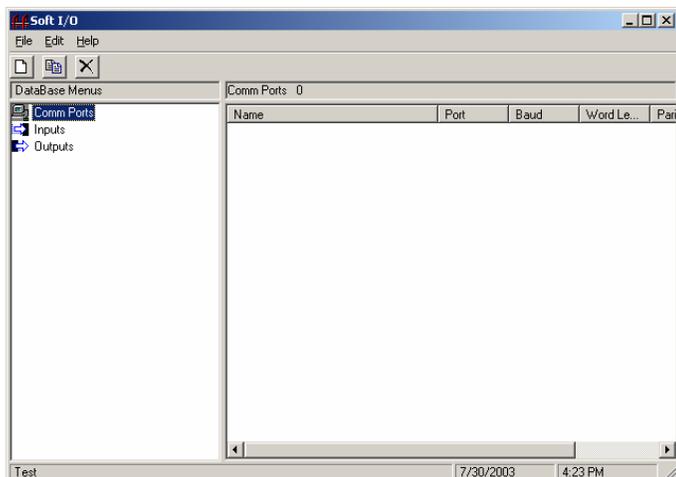
For Software upgrades it is necessary to uninstall the software from Control Panel > Add Remove Programs. Leave the directory in place and load the new software into the same directory.

- Follow the instructions and call DSX Technical Support to register the software. Be prepared to give the serial number as well as the registration number shown on the screen. DSX Technical Support will give you a return number to enter in the Key box. Each installation of the software must be registered. Multiple installations of the software can be used on the same PC.

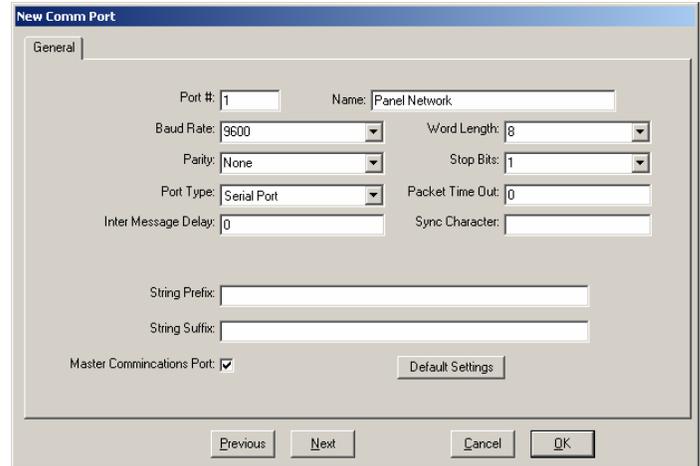


Configuring Soft IO

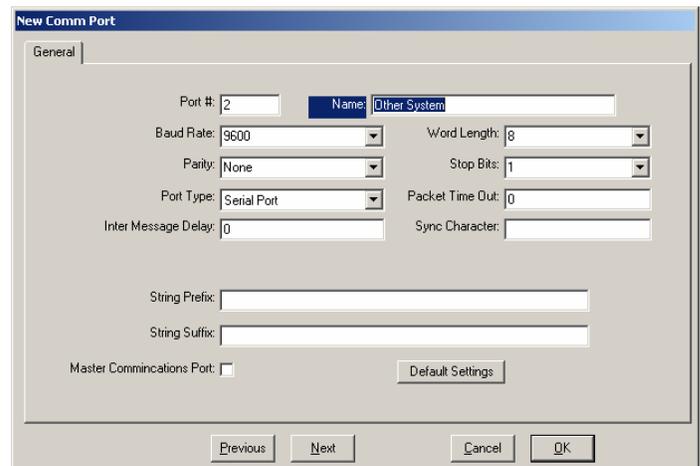
- Double click on the SioDB.exe again to launch the database portion of the program. This is where Soft IO is configured.
- The first item to be configured is Comm Ports. Two Com Ports are required at a minimum. One Comm Port is for communication with the DSX RS-485 controller network and the other is used to communicate with the other system. It is possible that we listen to the other system on one port and communicate back to it or to yet another system on a different Comm Port.
- Click on Comm Ports and then on the Add Button (blank page) in the tool bar. To utilize multiple serial ports it may be necessary to add hardware such as a Digi board to the computer.



- Enter the Communications Port Number and the Name and leave all other fields defaulted. At the bottom of the screen place a checkmark in the Master Communications Port selection box and then click on OK. This will be the Comm Port used to communicate with the RS-485 controller network.



- Click on Comm Ports again and then on the Add Button (blank page) in the tool bar. This is to Add another Comm Port that will be used to communicate with the other system. Set the Communication Parameters such as baud rate to match that of the other system. Do Not check the Master Communications Port selection box. Click OK to save. It may be necessary to define one port for listening to the other system (Inputs) and one for talking to the other system (Outputs). It may be we are listening to one system for Input and talking to another system for Output.



It is possible to communicate to the other system via TCP/IP. If this is the case set the Port # to a number that is out of the physical port range. Set the **Port Type** to TCP/IP and enter the **TCP/IP Address** and **IP Port#** of the other system. Use the Transmit IP Port # field to set the Port number we will be communicating to. If left to 0, SIO will use the next available number. Use the Receive IP Port # to set the Port number we will be listening on for communications. It is also possible for Soft IO to communicate with the DSX controller network via TCP/IP if the system is using "PC Master" software to communicate with slave controllers via IP.

String Prefix and **String Suffix** can be used to simplify the strings being transmitted to other systems. If the Data to be sent to the other system should always begin with and/or end with certain characters those characters can be placed in the Prefix and Suffix fields instead of placing them in the Output Open and Secure fields.

6. From the main menu select **Inputs** and then Click on the Add Button (blank page) in the tool bar.
7. **Name** the Input as to what it represents. It is a good practice to name it the same as you do in the WinDSX software.
8. Set the **Device Address** that Soft IO should represent. This Device must also be defined in WinDSX. Defining an Input and or Output instructs Soft IO to respond to Polls for that address from the Master Controller. A Soft IO device can be any address except 0 and 1. Soft IO can represent multiple addresses.

9. Set the **Input Number** that Soft IO should represent. This Input must also be defined in WinDSX. Soft IO can represent Inputs 1 – 16 for each device address.
10. Set the **Comm Port** that was defined for communication with the other system.

11. Enter the ASCII string of characters that represent the **Normal** state of this Input. When Soft IO receives this string it will inform WinDSX that this input is Normal. If the other system does not send an Input Normal string, check the box at the bottom labeled "Automatically Reset to Normal". This instructs Soft IO to reset the Input back to normal so it will be ready for another activation.
12. Enter the ASCII string of characters that represent the **Abnormal** state of this Input. When Soft IO receives this string it will inform WinDSX that this Input is Abnormal. Do the same for Input Trouble if Trouble conditions are to be used.
13. From the main menu select **Outputs** and then Click on the Add Button (blank page) in the tool bar.
14. **Name** the Output as to what it controls. It is a good practice to name it the same as you do in the WinDSX software.
15. Set the **Device Address** that Soft IO should represent. This Device must also be defined in WinDSX. Defining an Input and or Output instructs Soft IO to respond to Polls for that address from the Master Controller. A Soft IO device can be any address except 0 and 1. Soft IO can represent multiple addresses.

- Set the **Output Number** that Soft IO should represent. This Output must also be defined in WinDSX. Soft IO can represent Outputs 1 – 8 for each device address.
- Set the **Comm Port** that was defined for communication with the other system.

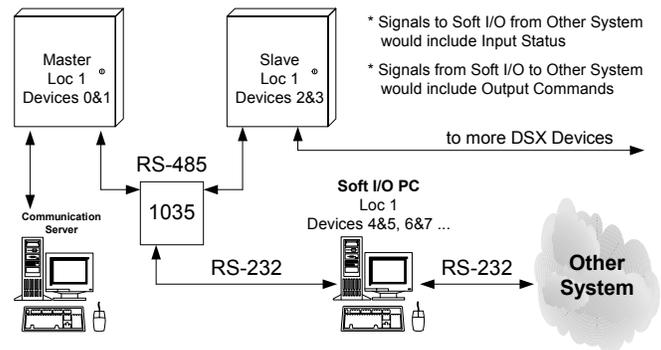
- Enter the ASCII string of characters that should be sent to the other system when the output goes **Open**. When the output goes Open for any reason (time zone, linking, manual commands, and scheduled overrides) Soft IO sends this string to the other system.
- Enter the ASCII string of characters that should be sent to the other system when the output goes **Secure**. When the output goes Secure for any reason (time zone, linking, manual commands, and scheduled overrides) Soft IO sends this string to the other system.

Configuring WinDSX for Soft IO

In WinDSX it is necessary to define the Devices, Inputs and/or Outputs that Soft IO should represent. These devices, inputs and outputs cannot be duplicated with actual panel addresses. The Master Controller will Poll for those addresses and the Soft IO computer will respond. As far as the Panel network is concerned those addresses are physically present. The WinDSX system and the controller network do not know about Soft IO and are programmed as usual. Inputs and Outputs that are used with Soft IO are programmed as normal with Time Zones, Action Messages, Linking and all other features.

Connecting Soft IO to Controller Network

Connect the Soft IO computer to the panel network similar to the drawing below. The DSX-1035 Quadruplexor installs anywhere after the Master Controller. Utilize an RS-485 Input for the upstream connection coming from the Master Controller. Use at least one RS-485 output for the downstream controller connections. Use the RS-232 output for connection to the Comm Port on the Soft IO computer. Be sure and cross transmit and receive between the 1035 and the Soft IO computer.



Questions and Answers about Soft IO

- If the Soft IO PC is turned off, is there an indication? Yes, if the Master Controller cannot communicate with Soft IO the WinDSX system will receive Device Communication Loss alarms and will display the Devices that Soft IO represents as offline.
- Can more than one copy of Soft IO be used on the same PC? Yes, although each would have to be purchased and registered separately they could be used to communicate with different DSX Controller Networks. For unidirectional Linking between two DSX Locations a single copy of Soft IO would be used. For bi-directional Linking between two DSX Locations two copies of Soft IO would have to be used.
- Can Soft IO communicate with more than one other system? Yes, a single copy of Soft IO will communicate with multiple other systems to receive input, and to multiple other systems to transmit output, within a single DSX Controller Network, all at the same time.