

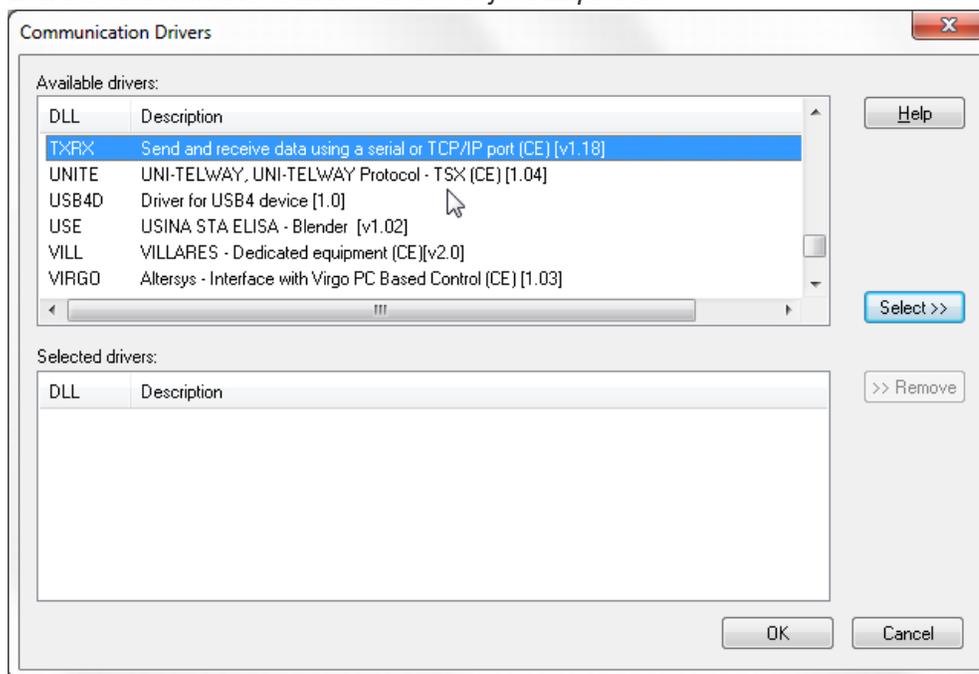
February 25, 2013

Convert from ASCII to Ethernet or Serial in just 15 minutes!!

The **TXRX driver** can be used to communicate between **InduSoft Studio** and other devices using Standard ASCII protocol with Serial (RS232) or Ethernet Interface. This is especially useful when working with devices that do not have their driver available. In this document, we will discuss how to install, configure and execute TXRX driver with InduSoft Web Studio v7.1.

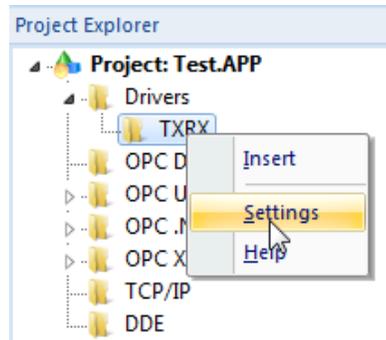
■ Installing the driver in InduSoft:

- From the main menu, go to *Insert > Add/Remove Driver*. It will open *Communication Drivers* window. Search for TXRX driver and click on *Select* to add it to the current project. It should be listed under *Selected Drivers* list. Click *OK* to close the dialog box and the driver is added to the *Drivers* folder in the *Comm* tab of *Project Explorer*.



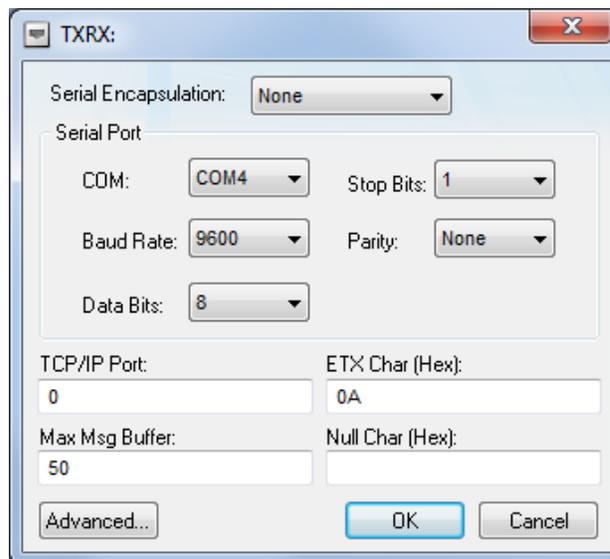
■ Configuring the Driver Settings:

- From the *Comm* tab on the *Project Explorer*, expand the *Drivers* folder and right click on TXRX, select *Settings*.



■ Serial Communication

- Communication Parameters dialog box for TXRX driver will be displayed and do the following settings:



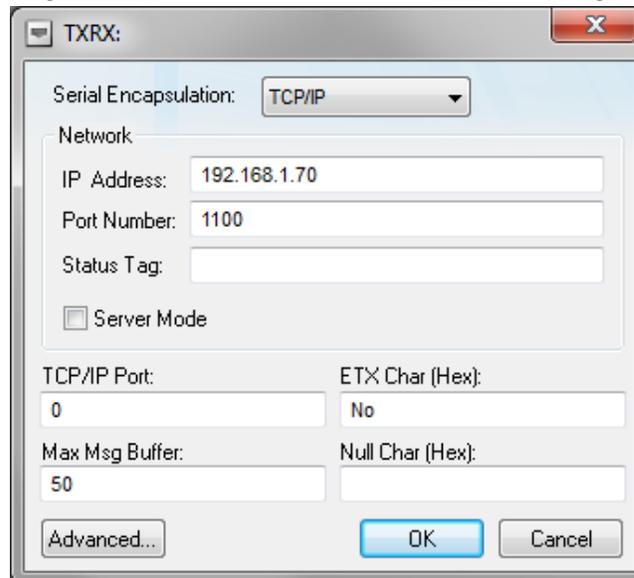
Note: Above is just an example. In practice, these settings should exactly match with the settings on your device to make correct communication possible.

- **Serial Encapsulation:** Encapsulation of serial data allows any serial device to communicate over an Ethernet network like any other TCP/IP network device. But when working with serial communication, select *None*.
- **COM:** Serial port of the PC used to communicate with the device
- **Baud Rate:** Data communication rate
- **Data Bits:** Specify number of data bits used in the protocol
- **Stop Bits:** Specify number of stop bits used in the protocol
- **Parity:** Specify parity of the protocol
- **TCP/IP Port:** Specify 0 for serial communication
- **ETX Char (Hex):** Used to indicate the end of message. For Read commands, InduSoft will identify end of message by this character. For Write commands, InduSoft will put this character at the end of message. In this example, line feed character 0A is used.
- **Max Msg Buffer:** Indicates the number of messages the driver will save in internal buffer.

- **Null Char (Hex):** When received message contains the specified 'Null Char' in it, the driver will duplicate the character so the user can distinguish between zero and 'Null Char'.
- **Advanced Settings:** If you are using any data converter (e.g. RS232 to RS485) between PC and target device, then you will have to configure these settings to account for the converter.

■ Ethernet Communication

- Do the following settings in the Communication Parameters dialog box:



Note: Above is just an example. In practice, these settings should exactly match with the settings on your device to make correct communication possible.

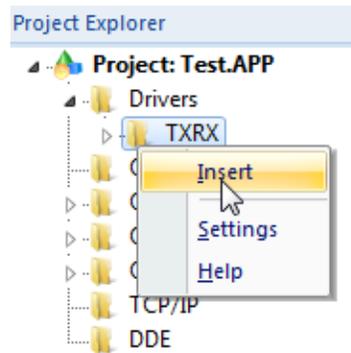
Serial Encapsulation: Depending on configuration of your target device, you may have to select Serial Encapsulation as *TCP/IP* or you might be able to communicate if you select *None* and provide the valid TCP/IP Port.

- If you chose *TCP/IP*, please specify the IP address, port number and other communication parameters like ETX Char, Max Msg Buffer and Null Char as shown above.
- If you do not chose *TCP/IP* and select *None*, ignore all the Serial Port (COM, Baud Rate, Data Bits, Stop Bits and Parity) fields. Specify the TCP/IP Port in syntax *<IP address>:<Port Number>* (e.g. 192.168.10.70:1100), ETX Char, Max Msg Buffer and Null Char.

■ Configuring the Driver Worksheet:

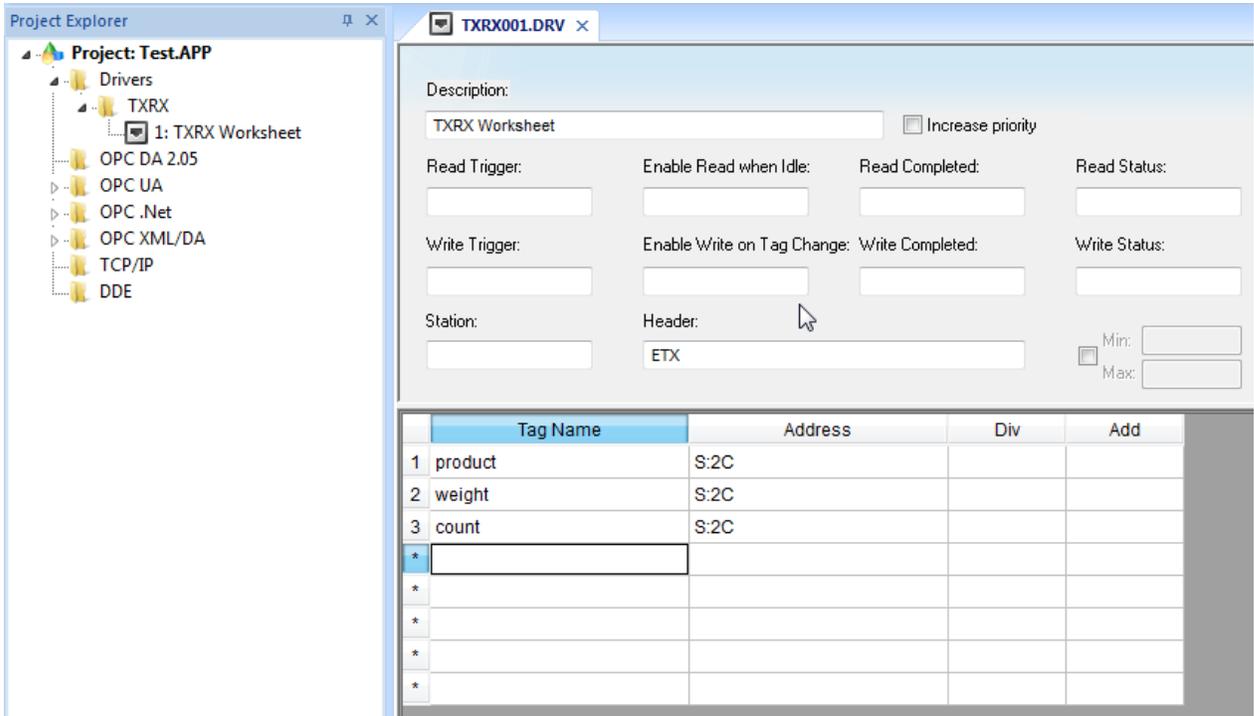
- Driver worksheets are used to associate InduSoft tag database to the parameters on the target device. We will discuss how to configure these worksheets to enable Read/Write operations with the target device. Whether you are working with Serial or Ethernet Interface, you will configure the same worksheet, but with different settings for different interfaces. You can also create more than one worksheet under TXRX driver.

- **Insert TXRX worksheet:** From the *Comm* tab on the *Project Explorer*, expand the *Drivers* folder and right click on *TXRX*, select *Insert* to add a worksheet.

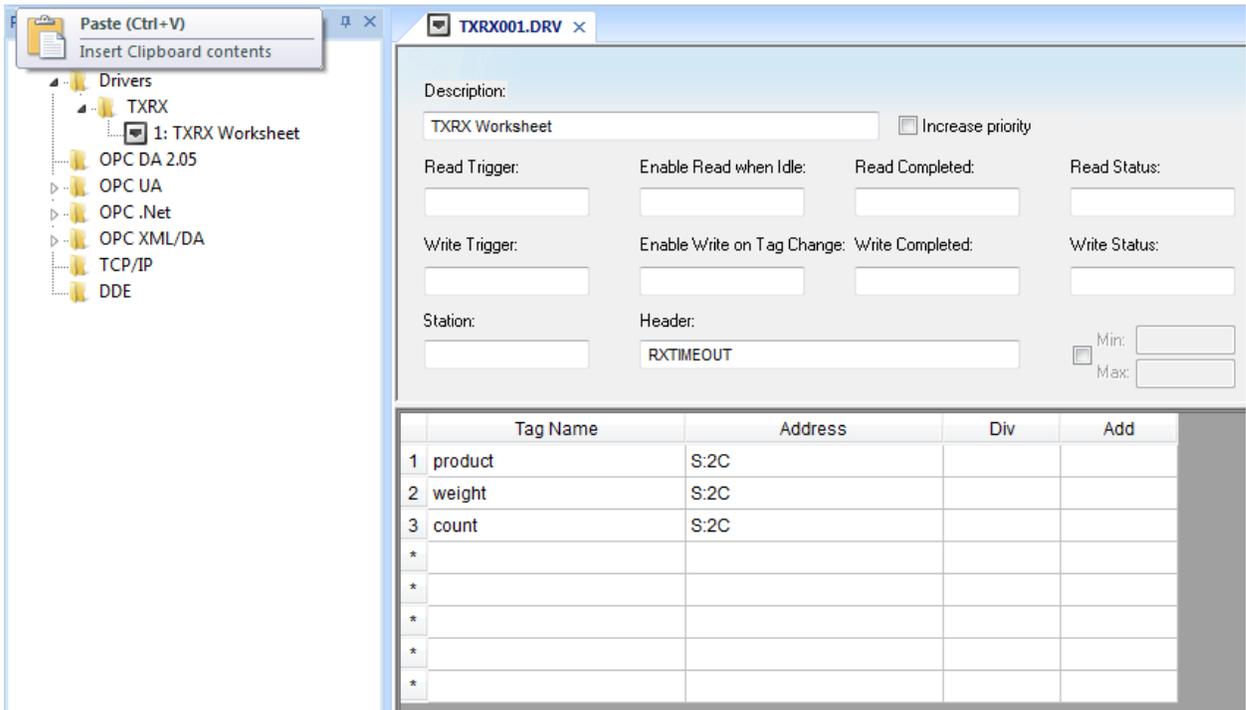


- A new TXRX worksheet will be inserted in the TXRX subfolder and it will be opened for configuration. Following are the configuration settings options:
 - **Description:** Give the description for the worksheet for documentation purpose
 - **Increase Priority:** Check this box to keep the commands on this worksheet on top of communication queue when triggered
 - **Read Trigger:** (Optional) Specify the tag that will trigger the read commands when the tag value is changed.
 - **Enable Read when Idle:** (Optional) Specify a tag or constant value (greater than 0) to enable reading from the device
 - **Read Completed:** (Optional) Specify a tag and driver will toggle its value when it completes a read command
 - **Read Status:** (Optional) Specify a tag and driver will update the tag with the status of last read command
 - **Write Trigger:** (Optional) Specify the tag that will trigger the write commands when the tag value is changed
 - **Enable Write on Tag Change:** (Optional) Specify a tag or constant value (greater than 0) to enable driver to check the worksheet continuously for change in tag values. If a change has occurred, the driver will write that value to the specified address
 - **Write Completed:** (Optional) Specify a tag and driver will toggle its value when it completes a write command
 - **Write Status:** (Optional) Specify a tag and driver will update the tag with the status of last write command.
 - **Station:** This field is not used for Serial communications, so you can leave it blank. When working with Ethernet communication, specify the IP address and port number in the following syntax: <IP address>:<Port Number>; Example: 192.168.10.70:1100
 - **Header:** Used to define specific incoming and outgoing messages. This field is used for both Serial and Ethernet communications.
 - Headers for incoming messages
 - **RXn** – writes incoming messages to specified tag when *n* characters arrive
Example: RX10 receives incoming messages of ten characters
 - **ETX** – writes incoming messages to specified tag only when given ETX character arrives
 - **RXTIMEOUT** – writes incoming messages to a specified tag only when there is no ETX character in the message

- Headers for outgoing messages
 - TX** – sends outgoing messages to the device; message consist of specified tag in ASCII with ETX character (syntax: *TX:<ETX(Hex)>*)
Example: TX:0A InduSoft uses 0A as ETX at the end of outgoing message
 - HTX** – message is written in Hexadecimal value (syntax: *TX:<ETX(Hex)>*)
Example: to write the value 10 in the tag, driver will send 0x10 to the device
 - **Min and Max:** Check to specify the minimum and maximum range of values and these values affect all tags in the worksheet.
 - **Tag Name:** Specify the tags for the values to be read/write to/from them.
 - **Address:** For each tag mentioned in the Tag Name column, configure the Address field using the following syntax:
 - For incoming and outgoing messages:
 - L:<Message Length>* - L:12 for tag to receive 12 characters
 - S:<Separator Char>* - S:2C for comma separated data (Separator Char specified in Hex)
 - S:<Any Integer Value>* - S:10 for tag to receive 10 characters
 - For Hexadecimal format output:
 - HL:<Message Length>* - HL:2
 - HS:<Separator Char>* - HS:32, Separator Char in Hex
 - For receiving values in Hexadecimal format, use *H*.
 - **Div:** Specify a division constant when scale adjustment is required. The project will use this value as a division factor for read and multiplication factor for a write operation. Do not use this if you have specified Min or Max values.
 - **Add:** Specify an addition constant when scale adjustment is required. The project will use this value as an addition factor for read and subtraction factor for a write operation. Do not use this if you have specified Min or Max values.
- Sample Worksheet for **Serial** Communication:

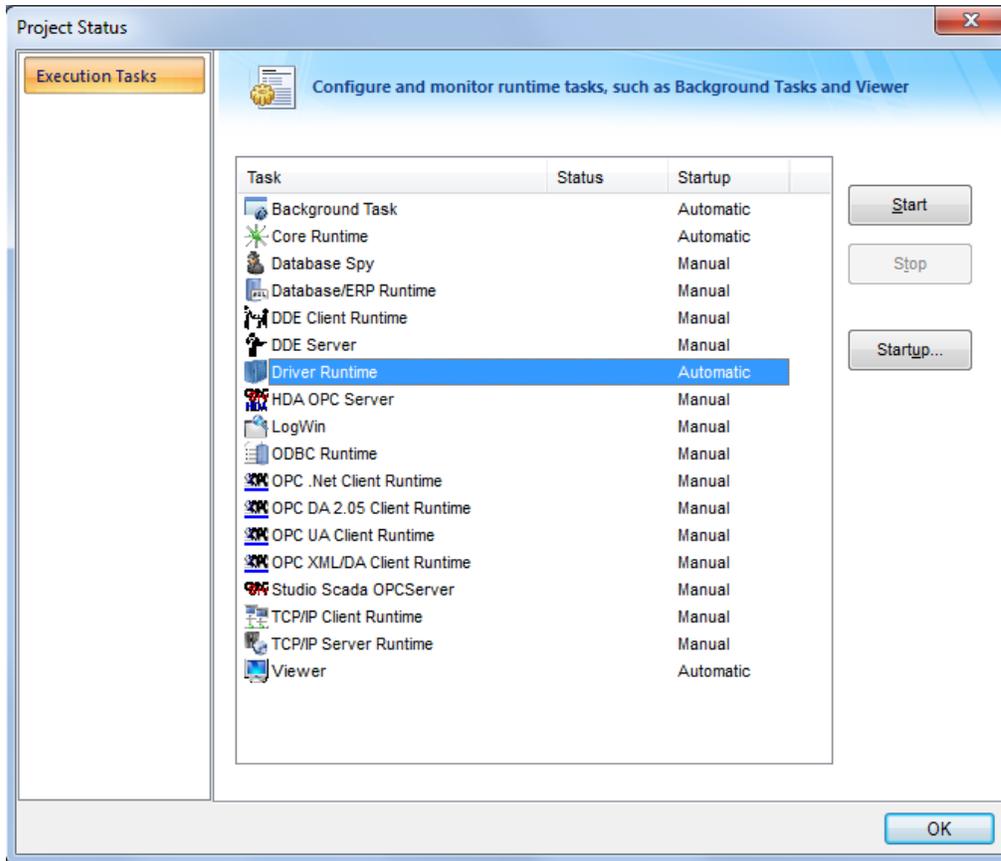


■ Sample Worksheet for **Ethernet** Communication:

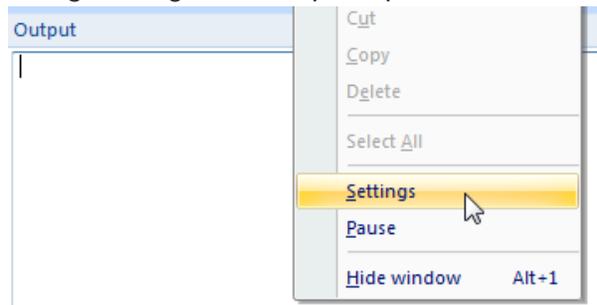


■ **Executing the Driver:**

- To verify your application's runtime execution with selected drivers, go to *Home > Tasks* (under local management area). A *Project Status* dialog box will open up.



- Verify that *Driver Runtime* is set to *Automatic*. If it is set to *Manual*, select the task and click *Startup* button to change it to *Automatic*.
- You can click on *Start* button to manually start the driver or click *OK* to close the dialog box and run the application.
- Also go to the tab *View* > check *Output Window*. An output window will be shown in the bottom right corner. Right click on the blank area of the output window and select *Settings*. A *Log Settings* dialog box will open up.



- Check the following: Field Read Commands, Field Write Commands, Insert Date/time and TCP/IP Messages (if required). You should be able to see the driver execution on the *Output Window*.

