

CTVUE Application Note

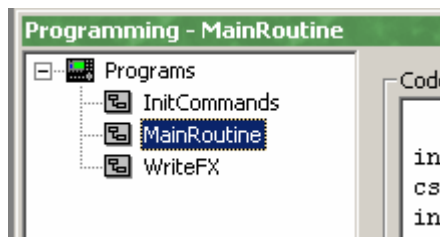
Number:CTVUE-1007, Revision 2, 10/24//2007

Subject: Reading/Writing to FX ASCII

Introduction

The CTVUE has the capability of configuring one of the serial ports to read/write ASCII data. A template program has been created that makes it easy to create tags that can read and write data to the FX drive. The following steps should be taken for a CTVUE to FX application.

Important Note: There are three programs in the CTVUE under the programs icon that are handling the ASCII communications and are linked to the tags you create and also the data you program on your Operator Interface. If you open the “programming” icon you will see the below programs.



The “Initcommands” program is used to set-up a value tag that specifies what command you want to send to the FX. The “mainroutine” program does the brunt of requesting and receiving data from the FX drive. The “writefx” program is pretty straight forward as it writes data from CTVUE to the FX drive.

Here the steps you need to take

- 1) Make RS-232 cable for connection from FX to CTVUE
- 2) Load FXASCIIstarter.ctv program into CTVUE
- 3) Setup FX serial command strings using FX manual: 40025500
- 4) Set-up commands strings
- 5) Create Tag Data
- 6) Create Data elements in the User Interface

CTVUE Application Note

Number:CTVUE-1007, Revision 2, 10/24//2007

Subject: Reading/Writing to FX ASCII

Step One RS-232 Pinout

FX

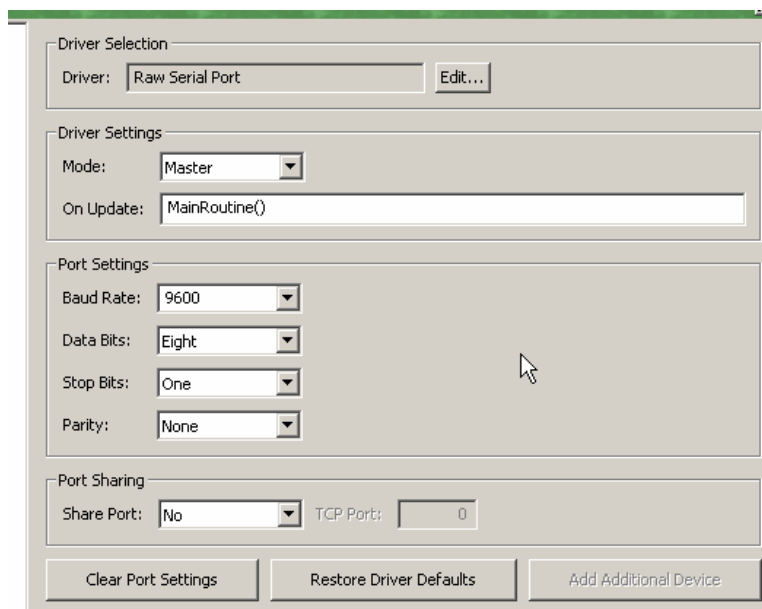
Drives : CTVUE-FX-232

Description: RS232 cable for connecting CTVue HMI to Epsilon EP or Unidrive SP

RJ11		9 pn subD	
CTVue pins		FX pins	
Rx	2	←————→	3 Tx
Tx	5	←————→	2 Rx
Common	3	←————→	5 0v

Step Two Load FXASCIIStartup.ctv program

Configure the RS-232 port to match the port you are using on the CTVUE and the baud rate settings on the FX drive. The FXASCIIStartup is setup to use the programming port on the CTVUE and for a baud rate of 9600 on the FX. Notice the program “MainRoutine()” is called from the raw serial port.



The screenshot shows a configuration window for the FXASCIIStartup program. It is divided into four sections: Driver Selection, Driver Settings, Port Settings, and Port Sharing. In the Driver Selection section, the 'Driver' is set to 'Raw Serial Port' with an 'Edit...' button. In the Driver Settings section, 'Mode' is set to 'Master' and 'On Update' is set to 'MainRoutine()'. In the Port Settings section, 'Baud Rate' is set to '9600', 'Data Bits' to 'Eight', 'Stop Bits' to 'One', and 'Parity' to 'None'. In the Port Sharing section, 'Share Port' is set to 'No' and 'TCP Port' is set to '0'. At the bottom, there are three buttons: 'Clear Port Settings', 'Restore Driver Defaults', and 'Add Additional Device'.

CTVUE Application Note

Number:CTVUE-1007, Revision 2, 10/24//2007
Subject: Reading/Writing to FX ASCII

Step Three Set-up FX Serial Command Strings

The FX uses two letter commands to handle various commands. For example the “DS” would be used for index position. The command to read index1 position would be “1DS1?” followed by a carriage return. The command to write to index1 position would be “1DS1= “value”. You will need to reference the FX manual to determine what commands are used for specific functions you need. Then you can use the tag editor to create tags for the various data and functions you want to perform.

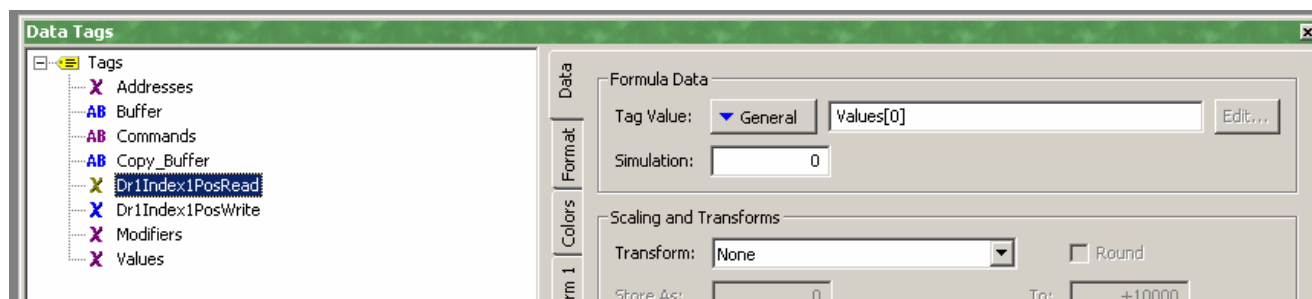
Once you have determined the command strings you will need to enter these into the “initcommands” program. There is a starter example for “1DS1” already in the program. It is broken down into three elements per the below graphic:

```
// This program is called on start up to fill an array with the command values.  
// Remember which element is used for each command, as you will need that for the value  
s  
  
// set element 0 for 1DS1  
Commands[0] = "DS";  
Modifiers[0] = 1;  
Addresses[0] = 1;
```

Commands are the main two letter command while Modifier is the number after the DS and the address is the number before the DS. You would add additional commands you may want to use and index the number in brackets. For Example:

Commands[1] =
Modifiers[1] =
Addresses[1] =

Then when you create a read tag you will reference this value in the tag setup per the below. Values is used in the MainRoutine to read the correct value.



CTVUE Application Note

Number:CTVUE-1007, Revision 2, 10/24//2007

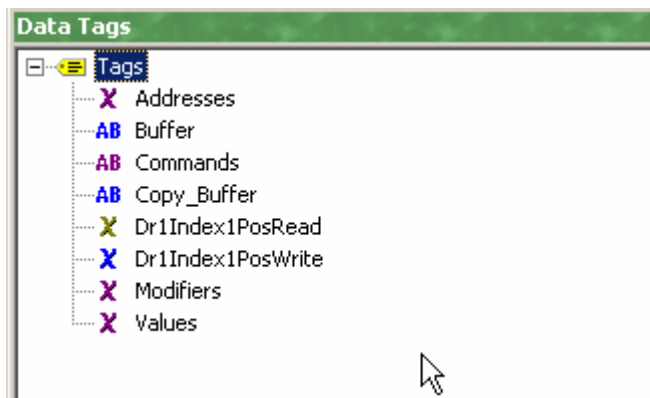
Subject: Reading/Writing to FX ASCII

Step Four Create Tag Data

When you open the Tag Editor you will see a read and write starter tag examples created and also some other internal and array tags used by the programs . Dr1Index1PosRead and Dr1Index1posWrite are the actual read and write data tags for exchanging information between CT-Vue and FX drive.

Read FX Data: Use Formula Tag Example: Dr1index1PosRead

Write FX Data: Use Integer Tag Example: Dr1IndexPosWrite



The Blue Buffer and Copy_Buffer tags are used in the programs to buffer incoming ASCII characters from the FX drive. The purple tags are array tags used with the initcommand and MainRoutine program to set-up the command strings. You reference the correct command by using the values[#] array where # is a number of the command you want.

CTVUE Application Note

Number:CTVUE-1007, Revision 2, 10/24//2007

Subject: Reading/Writing to FX ASCII

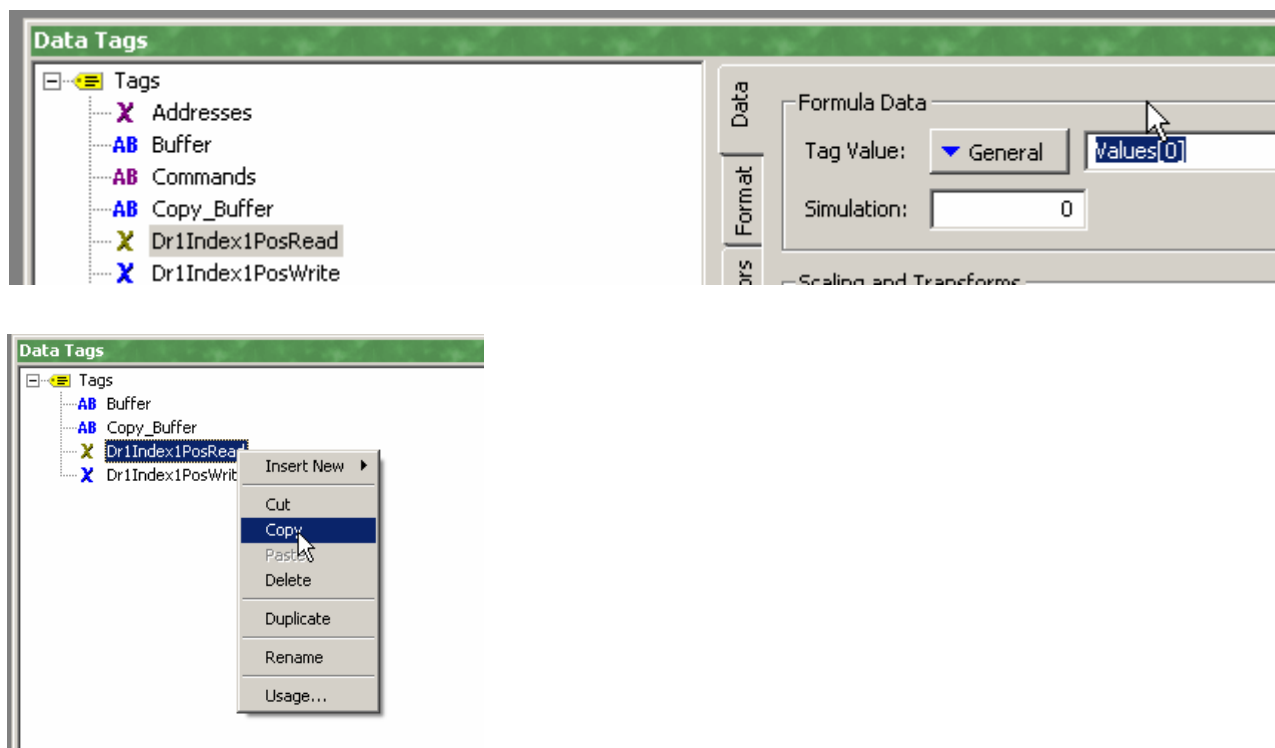
Creating a Formula Tag

The easiest way to create additional tags would be to just make copies of the existing tags. Remember you will need to index the value[#] if you want to read different data. The write command is done in the User Interface data entry properties.

Reading FX Values

The existing Dr1IndexPosRead is an integer formula tag. If you click on this tag you will see that it references an array tag called values[0]. Values[0] is where the read data is put when the program mainroutine runs. Program mainroutine runs constantly and is called under communications settings in the raw serial port settings.

So if you created a new tag and wanted to get different data then you would need to specify that data in the “initcommands” program as the next array element. Then you would create a tag and in the tag field you would put values[#] with # being the command you want to reference.



You have just configured a tag that will read data from the FX and convert to integer value.

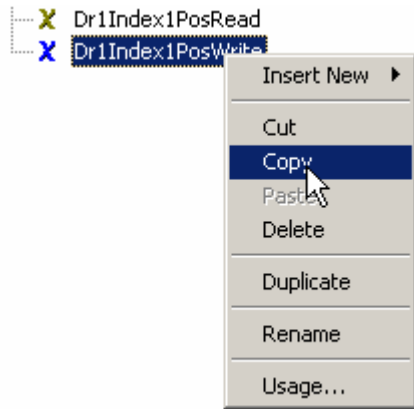
CTVUE Application Note

Number:CTVUE-1007, Revision 2, 10/24//2007

Subject: Reading/Writing to FX ASCII

Write data to FX

To write data to the FX you can make a copy of the example Dr1IndexPos1Write Tag. This tag type is a standard integer tag type which is converted to text using the write program. However, that is not done until you actual reference the tag within your user interface program.



CTVUE Application Note

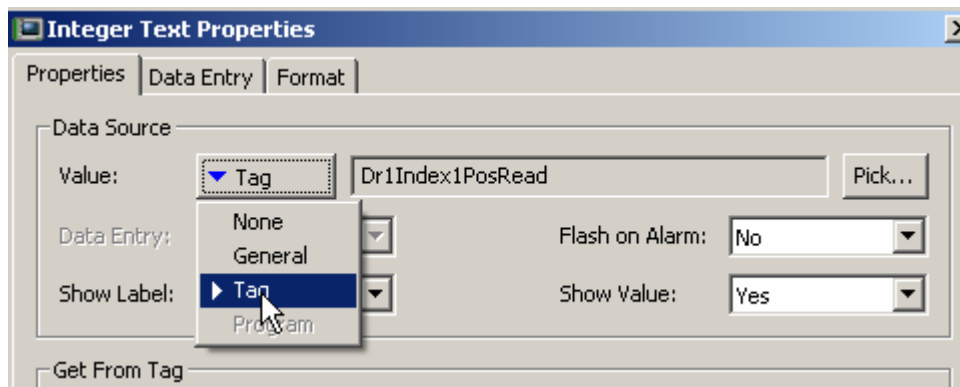
Number:CTVUE-1007, Revision 2, 10/24//2007
Subject: Reading/Writing to FX ASCII

Step Five Create Data elements in the User Interface

Once the tags have been created, you can begin to create the read/write data elements for your user interface.

Making Read Elements

Select the Integer Text Type to read data from the FX and drag it to your screen Then in the properties select the tag you want to read. That is all you have to do as the rest is accomplished from the tag properties you defined earlier.



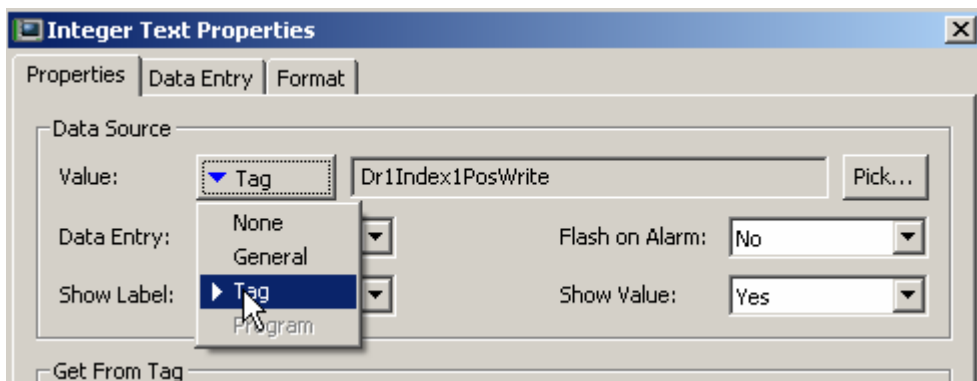
CTVUE Application Note

Number:CTVUE-1007, Revision 2, 10/24//2007

Subject: Reading/Writing to FX ASCII

Writing Data to the FX

Select the Integer Text Type to write data to the FX and drag it to your screen. Open the properties and select the tag you want to write to per the below. After this there is one more step for writing to the FX which requires an entry in the “Data Entry” tab of the properties.



CTVUE Application Note

Number:CTVUE-1007, Revision 2, 10/24//2007
Subject: Reading/Writing to FX ASCII

The data entry tab will be used to send the value to the FX drive as a text value and ensure the data is sent via the appropriate function. There are Five arguments you will have to enter in the WriteFX command as follows:

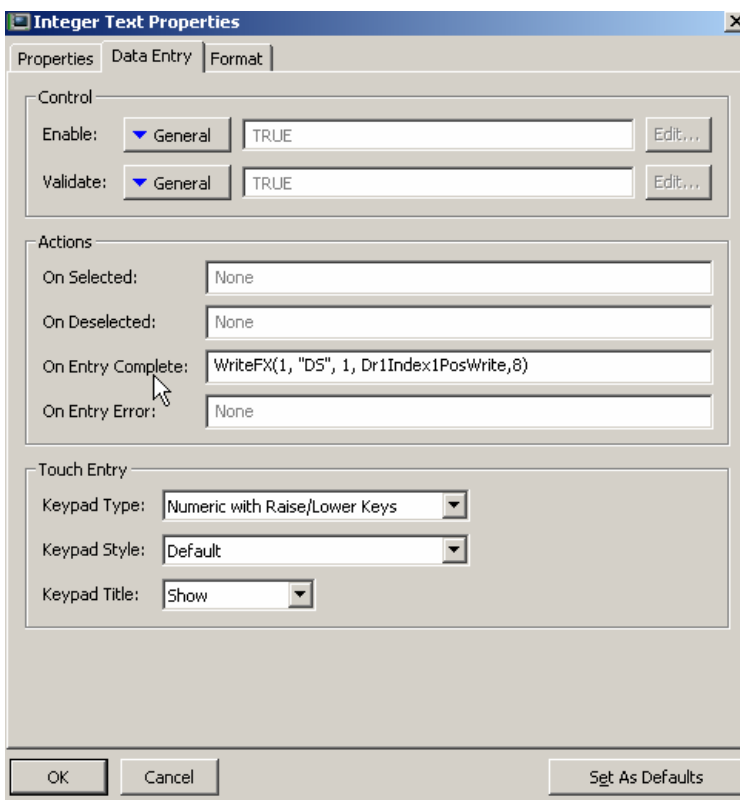
First argument: Drive Address

Second Argument: FX Two Letter Command

Third Argument: Actual Value that gets converted to text for sending to FX

Fourth Argument: Number of digits in the value

Note: Make sure your tag format number of digits matches the number of digits you are trying to write.



The screenshot shows the 'Integer Text Properties' dialog box with the 'Data Entry' tab selected. The 'Control' section has 'Enable' and 'Validate' both set to 'General' and 'TRUE'. The 'Actions' section has 'On Selected' and 'On Deselected' set to 'None', 'On Entry Complete' set to 'WriteFX(1, "D5", 1, Dr1Index1PosWrite,8)', and 'On Entry Error' set to 'None'. The 'Touch Entry' section has 'Keypad Type' set to 'Numeric with Raise/Lower Keys', 'Keypad Style' set to 'Default', and 'Keypad Title' set to 'Show'. The 'OK', 'Cancel', and 'Set As Defaults' buttons are at the bottom.

You have now setup a data element that will write data to the FX drive.