



TECHNICAL NOTE TNOI40

Title: Barcode Scanner Support

Product(s): G3, Modular Controller Enhanced Master & DSP

ABSTRACT

This document describes a simple application in which the user wants to receive data from a barcode scanner. For the purpose of this application, the barcode scanner is configured to automatically transmit the ASCII string upon a successful read, i.e. a “trigger” command is not being sent from the Red Lion device in order to request that the scanner send the string.

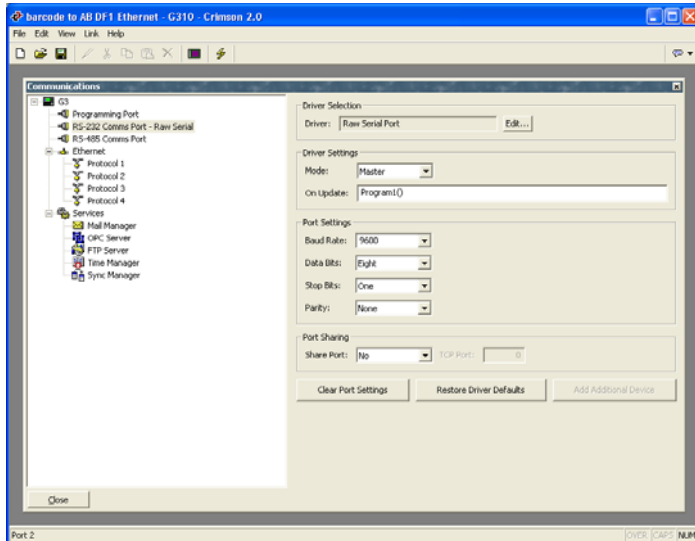
Any of the G3 series HMIs, Modular Controller (Enhanced Masters), and Data Station Plus series products may be used for this application.



INTRODUCTION

This example demonstrates how to use a G310 to receive data from an RS-232 equipped barcode scanner. The G310 will receive the ASCII string from the scanner, and store that string in a tag. That tag may then be used, as would any tag within Crimson.

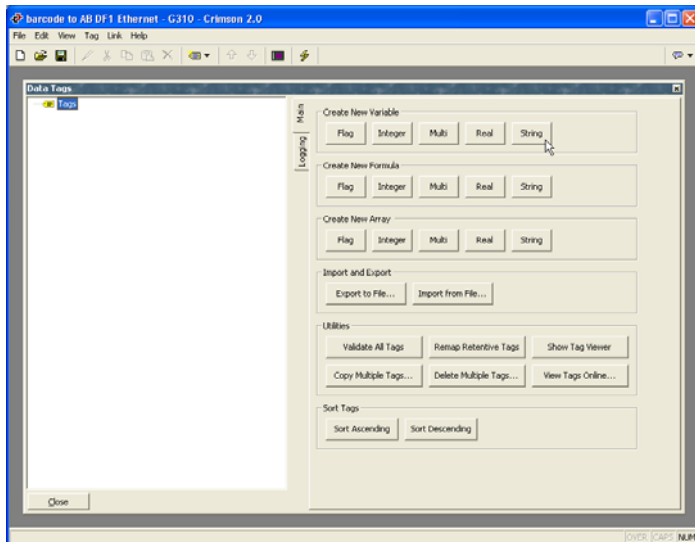
PORT CONFIGURATION



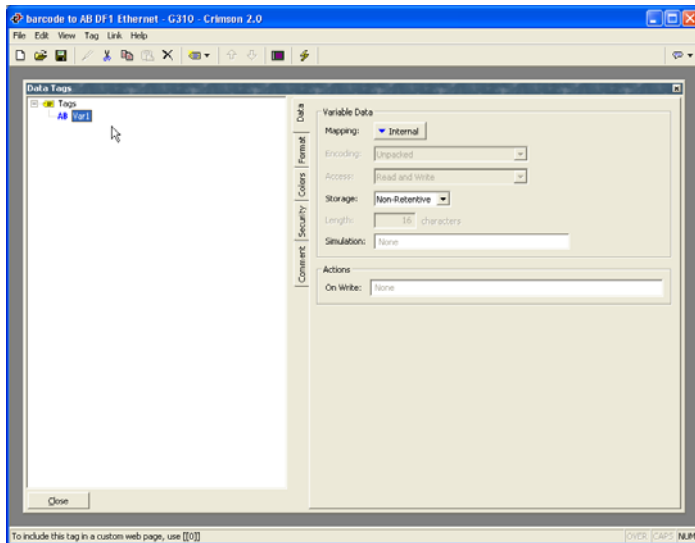
Choose the appropriate serial port for the barcode scanner, and select the Raw Serial port driver. Adjust the Port Settings to match those of the barcode scanner. Set the On Update property to `Program1()`. This will cause the G3 to call this program on every communications update.

Note: The task bar, located at the bottom of the pane, shows you the port number for the chosen serial port. This value (2 in this case) will be used within the program Reader.

TAGS

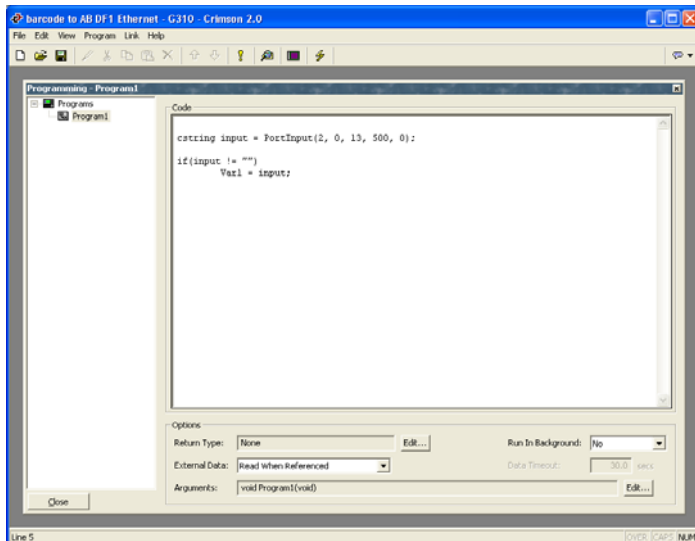


Go to the Tags section of Crimson. Create a string tag using the String button under the Create New Variable section.



For now, the default name Var1 is suitable, though you may choose to rename this tag to something more memorable. Further, you may want to adjust the Field Width of the tag under the Format Tab. This should be set to a value large enough to capture the string from the barcode scanner.

USER PROGRAM



Enter the following text into the Code area of Program1...

```
cstring input = PortInput(2, 0, 13, 500, 0);

if(input != "")
    Var1 = input;
```

Then click the yellow ! button on the toolbar, or press CTRL+T on your keyboard to translate, e.g. compile the code you've just entered.

EXPLANATION

```
cstring input = PortInput(2, 0, 13, 500, 0);
```

Cstring input creates a local variable called “Input”, and sets it equal to the contents of PortInput, which is the serial input buffer. The arguments, contained in parentheses, are as follows...

2 – The port from which the data is being received. This number is found on the task bar when configuring the port as mentioned previously.

0 – The character that the unit will look for to signify the start of the incoming string. Using 0 indicates no start character is defined.

13 – The character that the unit will look for to signify the end of the incoming string. In this case the string will end with a carriage return, which is 13 decimal.

500 – Delay in milliseconds that the unit will wait with no activity before processing the input buffer.

0 – The maximum number of characters to return. This, along with the end character, is what is used to signify that we have all of the string. Which ever happens first has priority. Leaving this at zero will return all characters received before the start character.

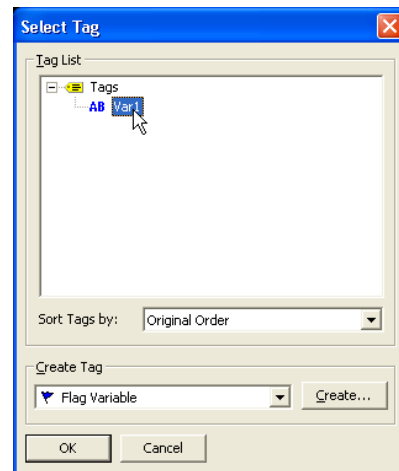
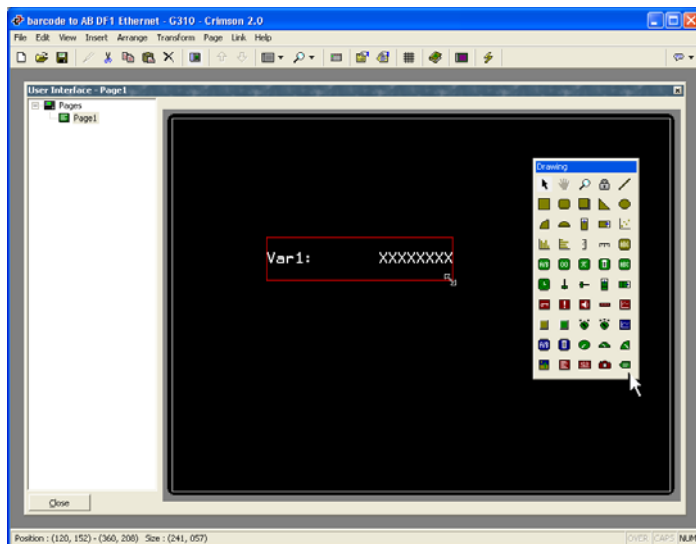
```
if(input != "")
```

If statement, which checks to see if, Input contains anything. “If the input is “not equal” to “” (empty), perform the action.

```
Var1 = input;
```

If Input is not empty, set Var1 equal to the contents of Input. This stores the barcode scanner’s string in the tag Var1.

USER INTERFACE



Click the tag button on the tool pallet to bring up the tag selection dialog. Select Var1, and click the OK button. On the page area, click and hold the mouse button, then drag the mouse to create the tag field. This field is where the barcode scanner’s output will be displayed.