



TECHNICAL NOTE TN0I38

Title: Connecting to Banner SureCross™ DX80 Wireless I/O

Product(s): G3, Data Station Plus and Modular Controller series

ABSTRACT

This document describes how to set up communications between Banner's SureCross™ DX80 Wireless I/O Gateway and Red Lion Controls G3 series of operator interfaces, Data Station Plus series or Modular Controller Series.

INTRODUCTION

This technical note will describe how to set up a G310 HMI to communicate with a DX80 Gateway via RS485 serial communication or a DX80 Gateway Pro via Ethernet. Using the G310 HMI, users can easily visualize and control any I/O on any node linked to the Gateway.

This document will also show how to use the HMI's powerful protocol converter to transfer data to virtually any communication enabled devices, using serial or Ethernet communication, as well as how to set up the embedded data logger.

TABLE OF CONTENTS

ABSTRACT	1
INTRODUCTION	1
TABLE OF CONTENTS.....	2
SETTING UP THE COMMUNICATION PORT	3
Using the DX80 Gateway RS485 Serial Communication Port.....	3
Using the DX80 Gateway Pro Ethernet Communication Port.....	5
ACCESSING AND FORMATTING DATA	6
Finding I/O Points Modbus Addresses	6
Digital I/O	7
Analog signals	10
Temperature inputs	12
Counter inputs.....	13
VISUALIZING DATA	14
TROUBLESHOOTING TOOLS.....	15
Checking Available Wireless Network Nodes.....	15
Performing a Site Survey.....	17
PROTOCOL CONVERSION.....	21
DATA LOGGING	24

SETTING UP THE COMMUNICATION PORT

USING THE DX80 GATEWAY RS485 SERIAL COMMUNICATION PORT

The Banner DX80 Gateway supports Modbus serial communication as standard and can be connected directly to the G310 HMI RS485 communication port. The G310 can then communicate and access any of Banner's DX80 I/O nodes (information and data) linked to the DX80 Gateway. The architecture would be as follows.



Figure 1

Please refer to the Banner documentation for installation instructions on the DX80 Gateway and I/O Nodes.

The cable wiring between the DX80 Gateway and the G310 RS485 communication port is described in the table below. The Red Lion side of the cable uses a standard RJ-45 jack for the RS485 signals. The Banner unit has a male M12 connector for both the RS485 signals and Power and Ground for the DX80 (Power on Pin 1 and Ground on Pin 3 of the M12 Connector). Care must be taken not to connect 24 VDC to the RS485 signal pins, as doing so could burn out the RS-485 circuits on both the Banner and Red Lion device.

Name	Connector Pinout	
	Red Lion RJ45	Banner Female M12
TxB	1	2
TxA	2	4
RxA	-	-
RxB	-	-
TxEN	-	-
COMM	6	5
TxB	-	-
TxA	-	-

From Crimson 2.0 main window, enter the Communications window.

1. Select the RS485 communication port
2. Click on the Edit button
3. Select Modbus under the Manufacturer
4. Select Modbus Universal Master as shown in Figure 2

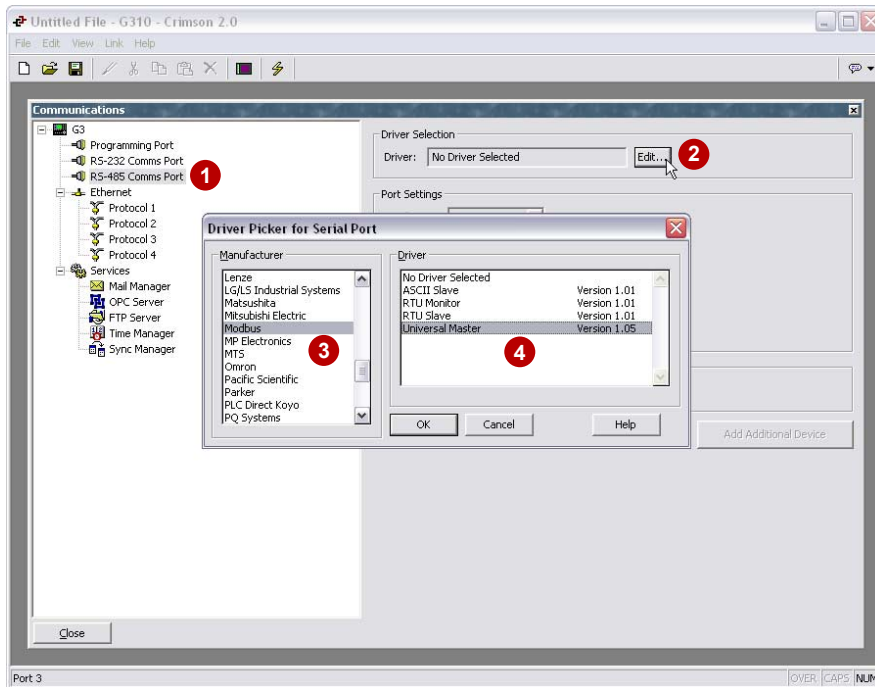


Figure 2

Change the Baud Rate to 19200 (default value of the DX80 Gateway) and then rename the device under the RS485 communication port to “GTW” (Figure 3).

Other Modbus settings correspond to the default communication settings in the DX80 Gateway. If the communication settings have been changed in the Gateway, please adapt the parameters in Crimson so they match.

The DX80 Gateway menu item called “SLID” stands for Modbus Slave ID, in Crimson this is the “drop number” parameter when the device is selected in the Communications window.

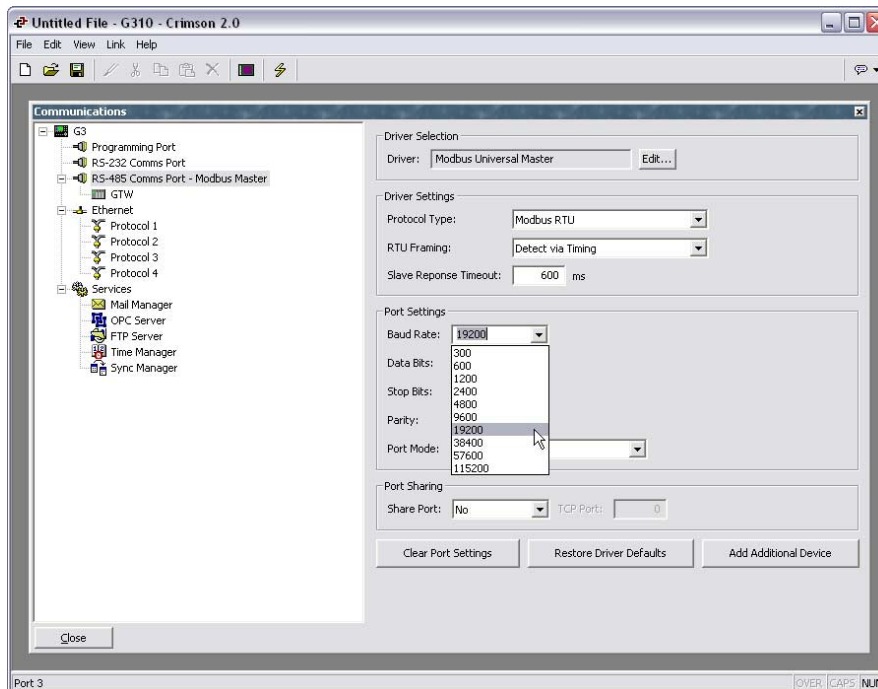


Figure 3

USING THE DX80 GATEWAY PRO ETHERNET COMMUNICATION PORT

The Banner DX80 Gateway Pro supports Ethernet Modbus TCP/IP communication and can be connected directly to the G310 HMI Ethernet port or via a hub, switch or network. The G310 can then communicate and access any Banner's DX80 I/O nodes (information and data) linked to the DX80 Gateway Pro. The architecture would be as shown in Figure 4.

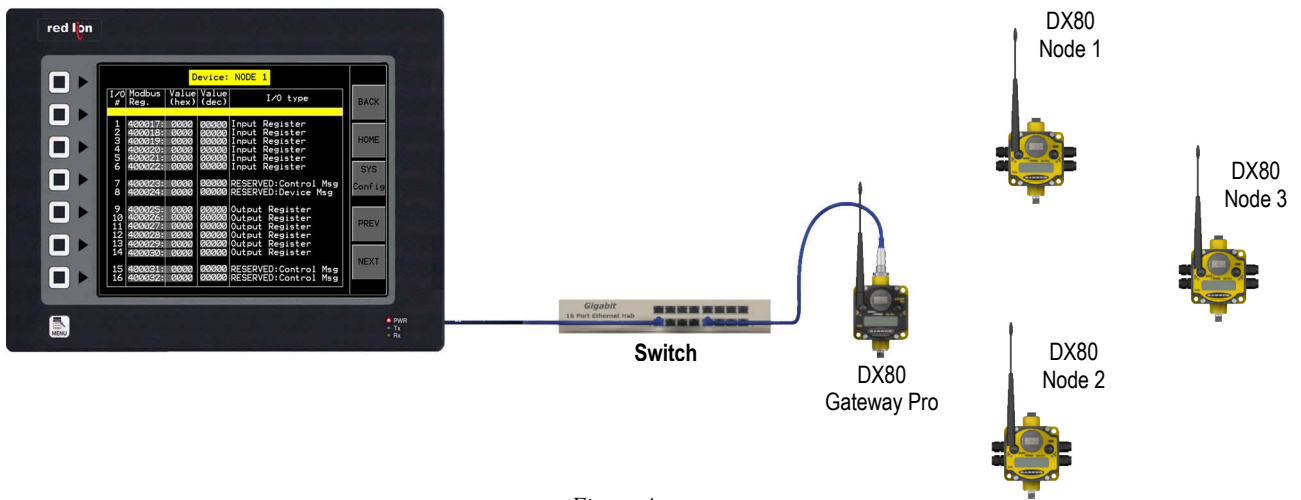


Figure 4

Please refer to the Banner documentation for installation instructions on the DX80 Gateway and I/O Nodes.

The DX80 Gateway Pro and the G310 can be connected directly using a standard Ethernet crossover cable or via a switch/hub using standard Ethernet straight cables.

In Crimson 2.0 Communications window:

1. Select the Ethernet port
2. Assign an IP address to the G310. In this example, we chose 192.168.0.5 (Figure 5). The Gateway Pro default IP address is 192.168.0.1.

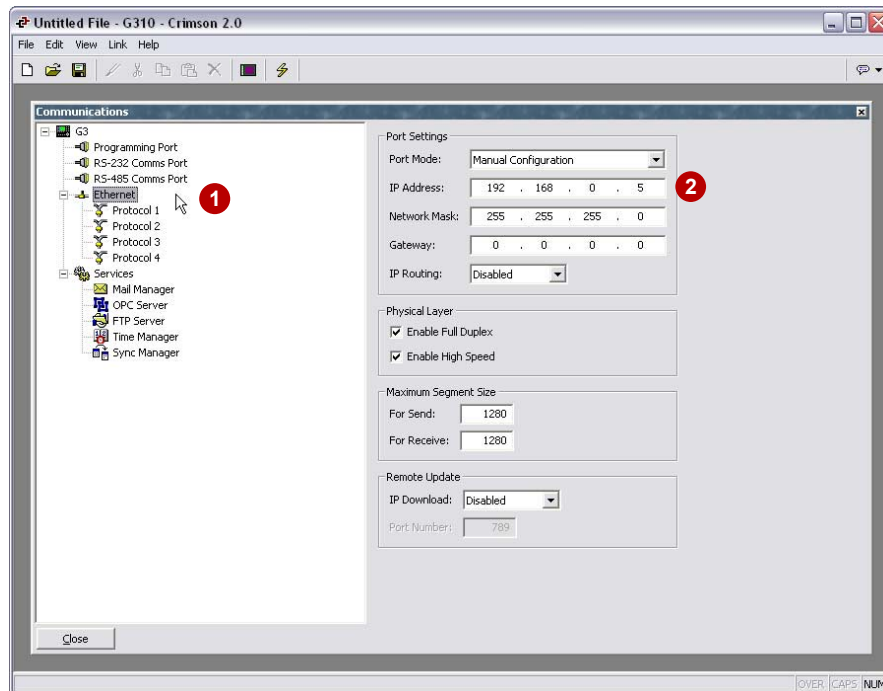


Figure 5

3. Select Protocol 1
4. Click the Edit button
5. Select Modbus under Manufacturer
6. Select TCP/IP Master driver as shown in Figure 6

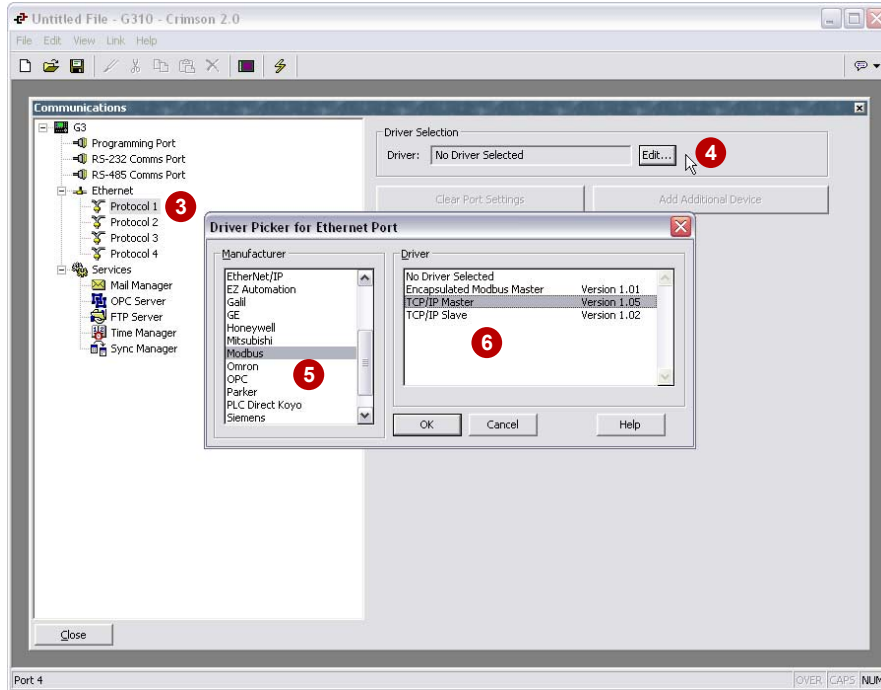


Figure 6

A new device PLC1 was created under Protocol 1. Rename it “GTW” and change the Device Identification IP address to the Gateway Pro IP Address, in this example 192.168.0.1 since we used the Gateway’s default settings (Figure 7).

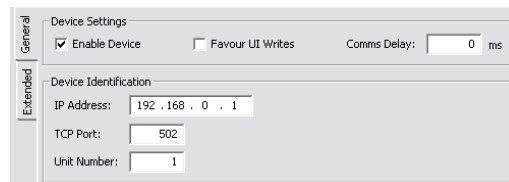


Figure 7

ACCESSING AND FORMATTING DATA

Once the communication is set up, data tags have to be created to access I/O information from the DX80 Gateway. Regardless of the communication type used, i.e. RS485 or Ethernet, tag creation, mapping and formatting will be the same.

FINDING I/O POINTS MODBUS ADDRESSES

All data are accessed via the DX80 Gateway using Holding Registers (4xxxxx). The targeted Modbus address defines which Node and I/O information is pulled. Let’s consider the following architecture, Figure 8; a Gateway and 3 nodes, addressed 1 through 3, with 6 digital inputs and 6 digital outputs each.

Note on addresses: Crimson 2.0 uses Modbus addresses with the format 400000. The Banner documentation uses the format 40000, with one less zero. Please note that although the Banner documentation references addresses in the shorter format, you should always use the longer address format in the Crimson software. For example, Crimson 400061 and Banner 40061 are the same address, holding register 61. The ‘4’ at the beginning informs that we are accessing Modbus holding registers.

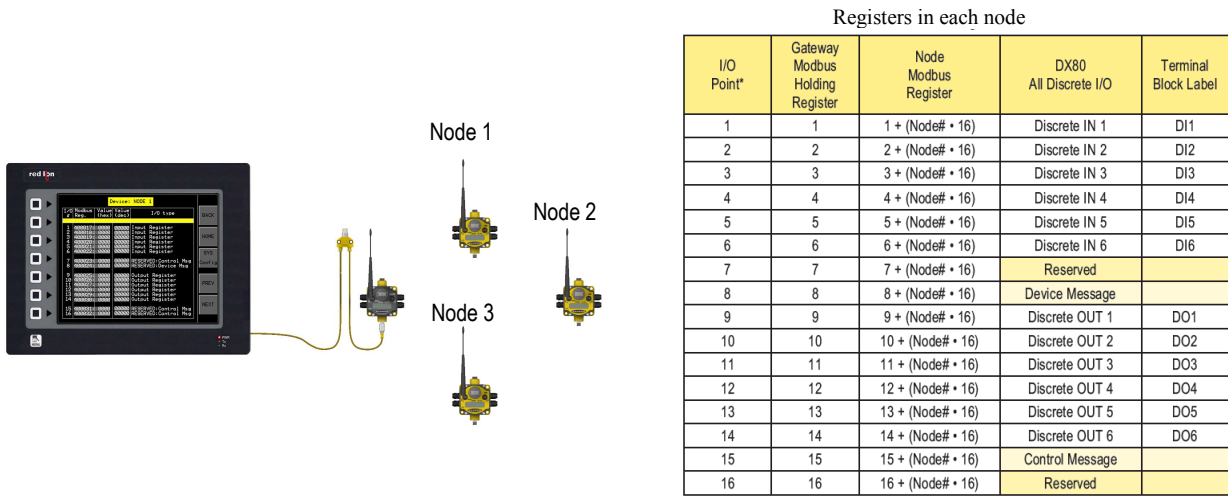


Figure 8

Each Node contains 16 Modbus registers (regardless of the node type). Above is a table example from Banner’s documentation for a Node with 6 digital inputs and 6 digital outputs. For a different Node type (analog, temperature, counter, etc.), please refer to Banner’s documentation for more details.

The “Node Modbus Register” column indicates the Modbus address formula to access the Node I/O point.

For example, to access input 3 in Node 1, the Modbus register to read will be:

$$\text{Node1_input3} = 3 + (1 \times 16) = 19, \text{ so } 40019$$

For example, to access output 5 in Node 3, the Modbus register to read/write will be:

$$\text{Node3_output5} = 13 + (3 \times 16) = 61, \text{ so } 40061$$

The DX80 Gateway can provide four different types of data depending of the Node type accessed; digital I/O, analog I/O, temperature and counter inputs. Each type has to be mapped differently and scaled to obtain the correct information in the data tag and on the display.

Note: Remember that Crimson uses the longer Modbus address format, so use Register number 400019 and 400061 in the Crimson software.

DIGITAL I/O

Depending on the Node model number used, there will be one input or output per register (as seen in the example above) or multiple inputs or outputs per register.

For example, model number DX80N9X6S6P6 will have one input or one output per register where DX80N9X6S8P4 will have 8 inputs on register 1 and one output per register.

The examples below do not refer to the architecture explained above, but to different wireless nodes type to illustrate the difference in data access.

Node with One Digital Input or One Digital Output per register

In Crimson 2.0 Data Tags, window, create a new Flag variable by clicking the Flag button in the Create New Variable category or via the menu Tag > Create > Variable > Flag. The new variable can be renamed to reflect the I/O point. In this example, it is rename Node1_Input1.

1. Select the newly created tag
2. Click on the Internal button next to Mapping. The GTW device created in Communications is available as shown in Figure 9.

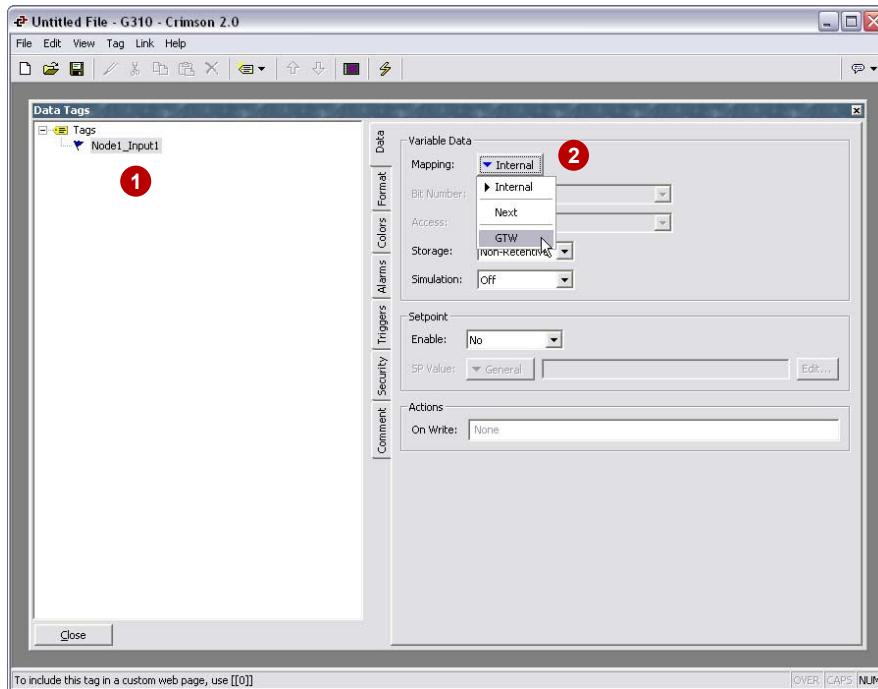


Figure 9

Click GTW and a window pops up to select the Modbus address to be accessed. In this example, Node1 Input would be $1 + (1 \times 16) = 17$, so Modbus address 400017.

Select Holding Registers for the Data Item and then enter 17 for the Element address (Figure 10). Then click OK.

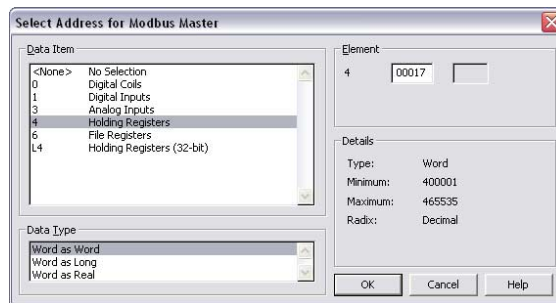


Figure 10

This tag is now linked to the gateway Modbus address 400017. **Bit Number 0** is shown under this address (Figure 11). Since the tag is a flag (A single bit type) but the Modbus address is a register (16 bits integer), Crimson has to know which bit should be targeted within the register. In our case Bit 0 is fine as the input state, on or off, will be in bit 0 of the register 400017.

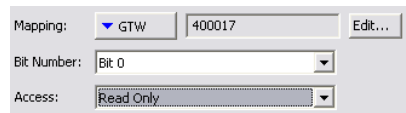


Figure 11

The **Access** parameter tells Crimson how the address can be accessed. In Read and Write mode, Crimson will allow reading and writing requests to be sent to this Gateway address. Since the address is an Input (Read only), this will result in a communication error in case of a write attempt. The Access parameter has to be changed to Read Only as shown in Figure 11.

If the tag is mapped to an Output type Modbus Address, then the access can be left as Read and Write.

Node with Multiple Inputs or Outputs per register

In Crimson 2.0 Data Tags, window, create a new Flag variable by clicking the Flag button in the Create New Variable category or via the menu Tag > Create > Variable > Flag. The new variable can be renamed to reflect the I/O point. In this example, it is renamed Node2_Input5.

1. Select the newly created tag
2. Click on the Internal button next to Mapping. The GTW device created in Communications is available as shown in Figure 12.

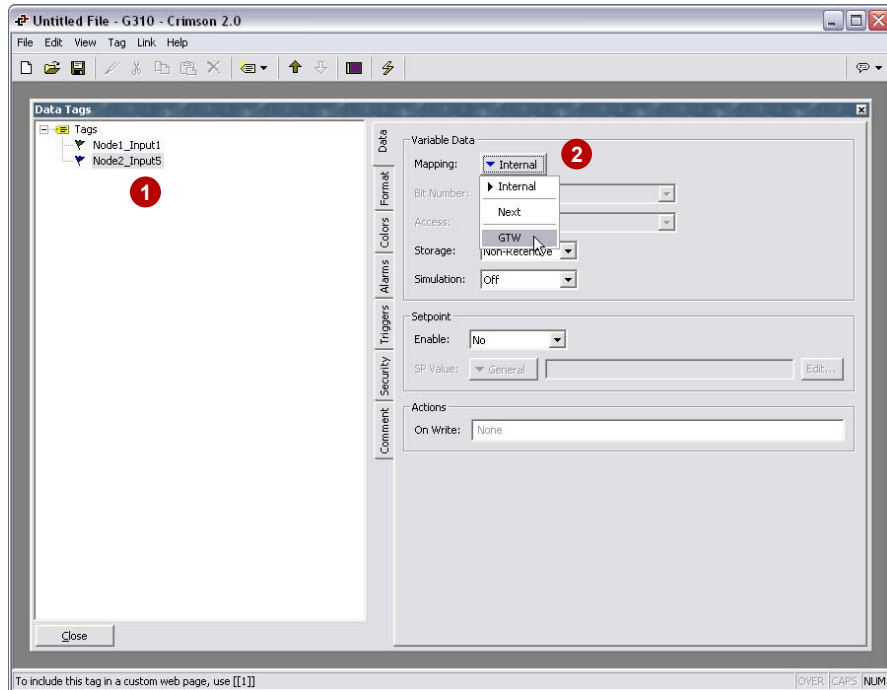


Figure 12

In this Node, all the inputs are in register 1. The bit number reflects the input monitored as shown in Figure 13.

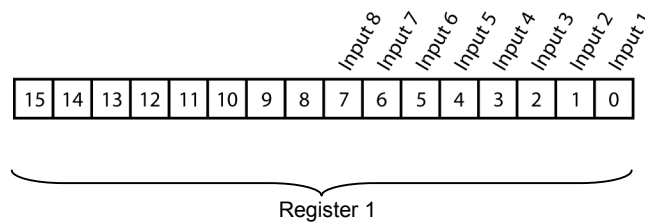


Figure 13

Click GTW and a window pops up to select the Modbus address to be accessed. In this example, Node2 Input5 would be $5 + (2 \times 16) = 37$, so Modbus address 400037.

Select Holding Registers for the Data Item and then enter 37 for the Element address (Figure 14). Then click OK.

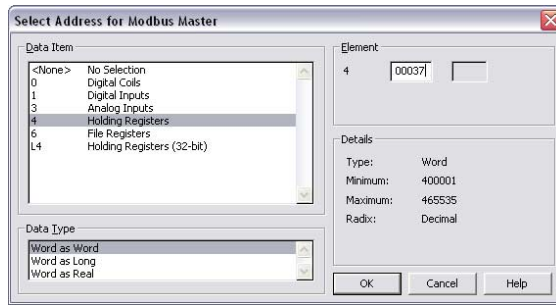


Figure 14

This tag is now linked to the gateway Modbus address 400037. This address contains the statuses of all the inputs. By default Crimson looks at Bit 0, which is Input 1 (as shown in figure 13). Changing the **Bit Number** parameter to Bit 4 will access Input 5 status (Figure 15).

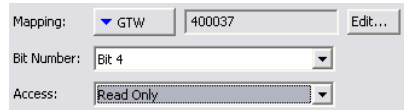


Figure 15

The **Access** parameter tells Crimson how the address can be accessed. In Read and Write mode, Crimson will allow reading and writing requests to be sent to this Gateway address. Since the address is an Input (Read only), this will result in a communication error in case of a write attempt. The Access parameter has to be changed to Read Only as shown in Figure 15.

If the tag is mapped to an Output type Modbus Address, then the access can be left as Read and Write.

ANALOG SIGNALS

In Crimson 2.0 Data Tags window, create a new Integer variable by clicking the Integer button in the Create New Variable category or via the menu Tag > Create > Variable > Integer. The new variable can be renamed to reflect the I/O point. In this example, it is rename Node3_Input1.

1. Select the newly created tag
2. Click on the Internal button next to Mapping. The GTW device created in Communications is available (See Figure 12 and 14 for similar screen references).

Click GTW and select the Modbus address to be accessed. In this example, Node3 Input1 would be $1 + (3 \times 16) = 49$, so Modbus address 400049.

Select Holding Registers for the Data Item and then enter 49 for the Element address. Then click OK.

This tag is now linked to the gateway Modbus address 400019. The value in the register is a raw unsigned integer, and will provide a range of 0 to 65535 for the analog input full scale as shown in the table below.

	INPUT 0 – 20 MA		INPUT 4 – 20 MA		INPUT 0 – 10V	
	Min	Max	Min	Max	Min	Max
Input	0	20	4	20	0	10
Register	0	65535	0	65535	0	65535

For easier reading, the register data can be scaled to the analog input process scale. For example, if a 4-20 mA input represent a tank volume from 0 to 1000 gallons. The register and display value would be as follows.

INPUT 4 – 20 MA		
	Min	Max
Input	4	20
Register	0	65535
Display	0	1000

This can be achieved in Crimson using linear scaling (Figure 16).

1. Select the Analog Input tag
2. Change the Transform property to Linear Scaling
3. Enter the scaling data:
 - Store As: 0 To: 65535
 - Display As: 0 To: 1000

The **Store As/To** property is the register memory range, i.e. the value read by the G3 from Modbus.

The **Display As/To** property is the respective value displayed for the **Store As/To** register values.

If the input is 4 mA, the register value would be 0 and the display will show 0. If the input is 12 mA, the register value would be 32767 and the display will show 500.

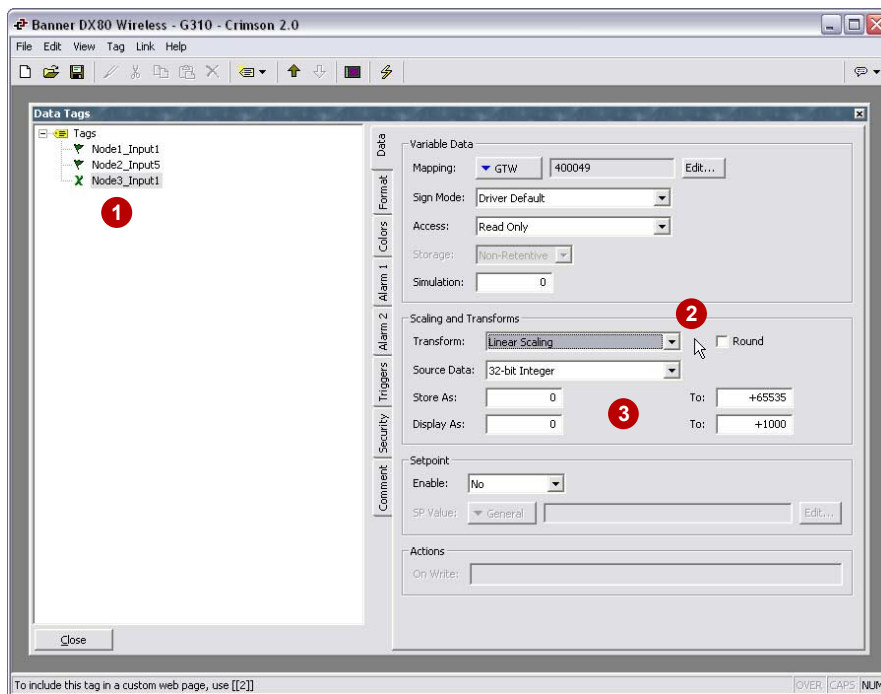


Figure 16

Note: Since analog tags are integers, scaling does not take the tag format. In the above example, if the tag is formatted for four digits before the decimal point and one digit after the decimal point, the display will show 100.0 for a 20 mA signal. In order to show 1000.0, the Display To value should be 10000.

Note 2: Make sure the access parameter is set to Read Only as the address is an input value.

TEMPERATURE INPUTS

In Crimson 2.0 Data Tags, window, create a new Integer variable by clicking the Integer button in the Create New Variable category or via the menu Tag > Create > Variable > Integer. The new variable can be renamed to reflect the I/O point. In this example, it is rename Node4_Input2.

1. Select the newly created tag
2. Click on the Internal button next to Mapping. The GTW device created in Communications is available (See Figure 12 and 14 for similar screen references).

Click GTW and select the Modbus address to be accessed. In this example, Node4 Input2 would be $2 + (4 \times 16) = 66$, so Modbus address 400066.

Select Holding Registers for the Data Item and then enter 66 for the Element address. Then click OK.

This tag is now linked to the gateway Modbus address 400066. The value in the register is the temperature times twenty ($T \times 20$). The temperature is in degree F or C depending of the sensor configuration, F is the factory default. The same scaling technique as the Analog inputs can be used (Figure 17), however only two scaling sets are possible, one to display whole degrees (register / 20), the other for tenth of a degree (register / 2).

The table below shows the Store As/To values for linear scaling on the tag.

	WHOLE DEGREE		TENTH DEGREE	
	As	To	As	To
Store	0	65535	0	65535
Display	0	3277	0	3277

3. Enter the scaling data

Temperature inputs are signed integers and thus require some extra formatting to display correctly.

4. On the Data tab, set the Sign Mode to Treat as Signed
5. On the Format tab, set the Sign Mode to Soft Sign.

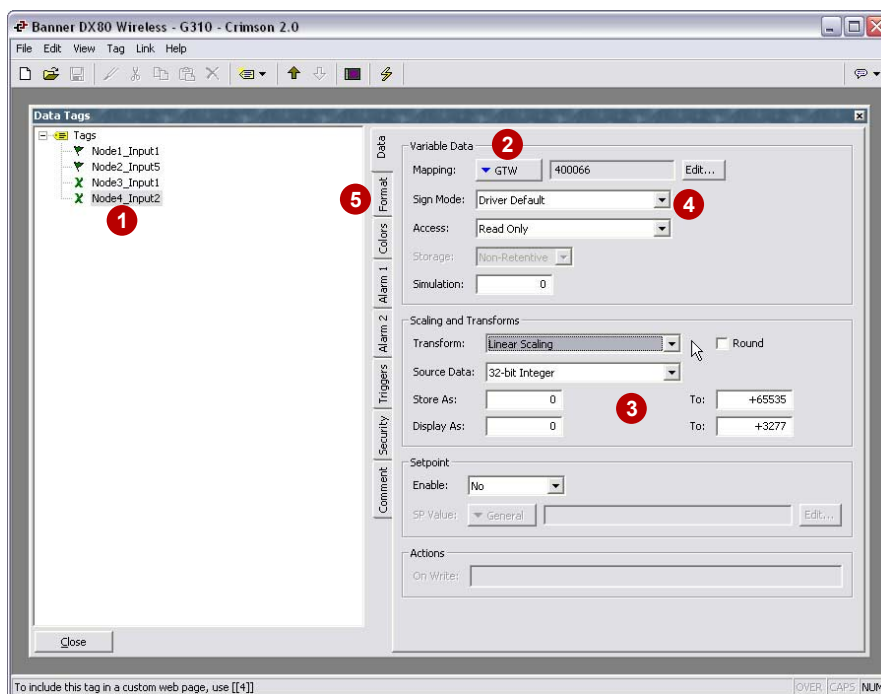


Figure 17

Note: Since temperature tags are integers, scaling does not take the tag format in account. Use no digit after the decimal point for whole degrees scaling values and one digit after the decimal point for tenth of a degree scaling values. Do not try to display hundredth of a degree accuracy, the value transmitted from the node does not have enough accuracy to justify this number of significant figures.

If the scaling for tenth of a degree is used, change the Digits before DP setting to 4 and the Digits after DP to 1 in the Format tab. This will add the decimal point where it belongs.

Note 2: Make sure the access parameter is set to Read Only as the address is an input value.

COUNTER INPUTS

In Crimson 2.0 Data Tags window, create a new Integer variable by clicking the Integer button in the Create New Variable category or via the menu Tag > Create > Variable > Integer. The new variable can be renamed to reflect the I/O point. In this example, it is rename Node5_Counter1.

1. Select the newly created tag
2. Click on the Internal button next to Mapping. The GTW device created in Communications is available (See Figure 12 and 14 for similar screen references).

The table in Figure 18 shows the available counter inputs for a DX80 Node (model DX80N9X2S4A2). Counter 1 uses two words, a high word in address 3 and a low word in address 4. The full counter value is therefore a double word, or long variable.

To map this variable in Crimson,

3. Click GTW and select Holding registers
4. Select Word as Long. This tells Crimson that the variable is a long register, so it will access the given Modbus address and the consecutive one
5. Enter the first register Modbus address. In this example, Node5 Counter1 would be $3 + (5 \times 16) = 83$, so Modbus address 400083.

I/O Point*	Gateway Modbus Holding Register	Node Modbus Register	DX80 All Discrete I/O	Terminal Block Label
1	1	1 + (Node# * 16)	Discrete IN 1	DI1
2	2	2 + (Node# * 16)	Discrete IN 2	DI2
3	3	3 + (Node# * 16)	Event Count IN 1 High Word	CI1
4	4	4 + (Node# * 16)	Event Count IN 1 Low Word	CI1
5	5	5 + (Node# * 16)	Count IN 2 High Word**	CI2
6	6	6 + (Node# * 16)	Count IN 2 Low Word**	CI2
7	7	7 + (Node# * 16)	Reserved	
8	8	8 + (Node# * 16)	Device Message	
9	9	9 + (Node# * 16)	Discrete OUT 1	DO1
10	10	10 + (Node# * 16)	Discrete OUT 2	DO2
11	11	11 + (Node# * 16)	Clear Counter 1	
12	12	12 + (Node# * 16)		
13	13	13 + (Node# * 16)	Clear Counter 2	
14	14	14 + (Node# * 16)		
15	15	15 + (Node# * 16)	Control Message	
16	16	16 + (Node# * 16)	Reserved	

Figure 18

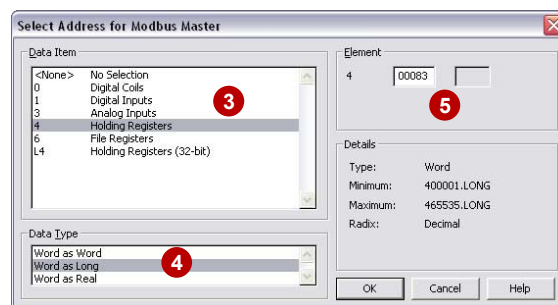


Figure 19

Crimson will then access 400083 and 400084 and provide the value of the counter in the tag Node5_Counter1.

Note: Make sure the access parameter is set to Read Only as the address is an input value.

VISUALIZING DATA

Once the Data Tags are created, enter the User Interface window in Crimson. Click in the display area to activate the drawing toolbox.

1. Click on the tag symbol to open the tag selection window (Figure 19)
2. Select the tag to display and click OK
3. Click and drag the mouse on the display until the tag shows up (Figure 20).

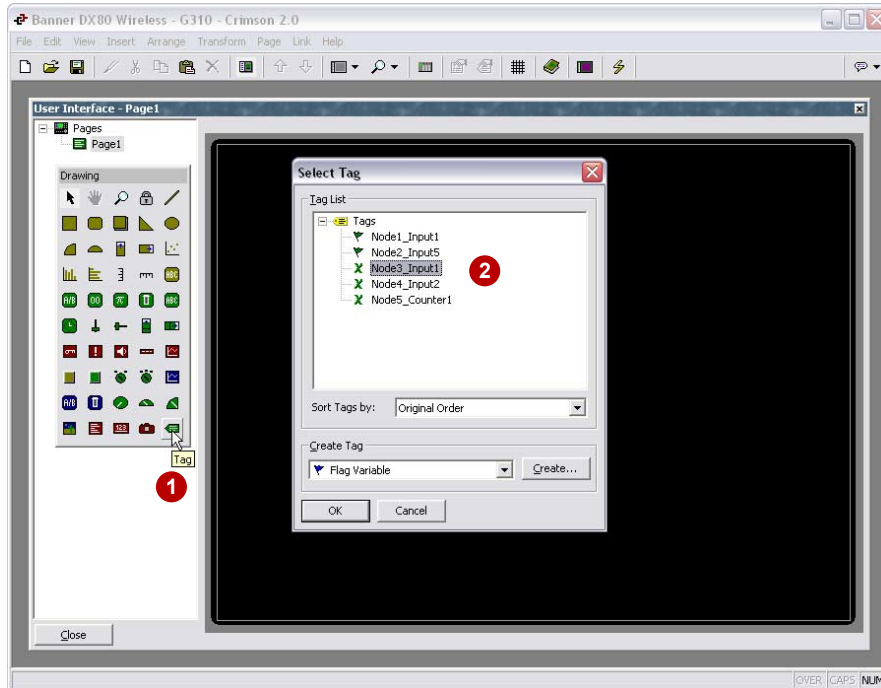


Figure 19

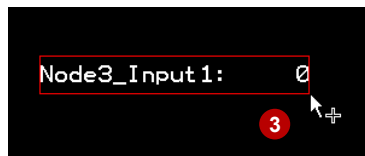


Figure 20

The format and scaling were already done at the tag level, so no extra operations are required.

TROUBLESHOOTING TOOLS

CHECKING AVAILABLE WIRELESS NETWORK NODES

This section demonstrates how to access Node availability information from the DX80 Gateway.

The Node Modbus register 8 is reserved for device messages or Site Survey data (when in Site Survey mode). The Node checks for temperature and battery problems before every transmission back to the Gateway. Other conditions are detected as they occur and are immediately reported back to the Gateway. Once the error message is sent back to the Gateway, the Node does not resend the message until the error condition changes or there is a higher priority message. The higher the message code, the higher the priority.

The Modbus register 8 is as follow,

	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Device Reg 8	Message Code								Data Field							

I/O 8 Device Messages		
Message Code	Data Field	Description
00	128	Normal Operation - 128 in the data field indicates a device is synchronized with the Gateway, 0 indicates no device is present.
		Warning Conditions
01	Message	Unknown message - Message was received correctly (correct checksum), but is not a known command.
		Error Conditions
53		RF Device timeout - A Node is not responding. The defined polling interval with allowable missed count was reached.
54		Modbus timeout - A Gateway Modbus timeout (time of inactivity on the serial channel) was detected.
254		Modbus register 8 device messages are disabled. The Modbus register 8 clears or disables messages using the Gateway's Modbus register 15.

* Modbus errors or warnings are indicated on the front panel LCD on the Gateway.

So, for example, to get Node 1 status, register 400024 (8 + (1x16)) has to be accessed.

There are multiple ways to show node availability to the user in Crimson, this example will use color codes for the three major states that can be extracted from the table above:

- Register value is 0 – The node has not yet joined the wireless network – Yellow
- Register value is 128 – Normal operation – Green
- Register value above 128 – The node is in error – Red

In Crimson, map an Integer tag to the Node message register address, in this example 400024 for Node 1.

1. Select the tag
2. Click on the Colors tab
3. Enter the limit value and change the foreground as shown in Figure 21.

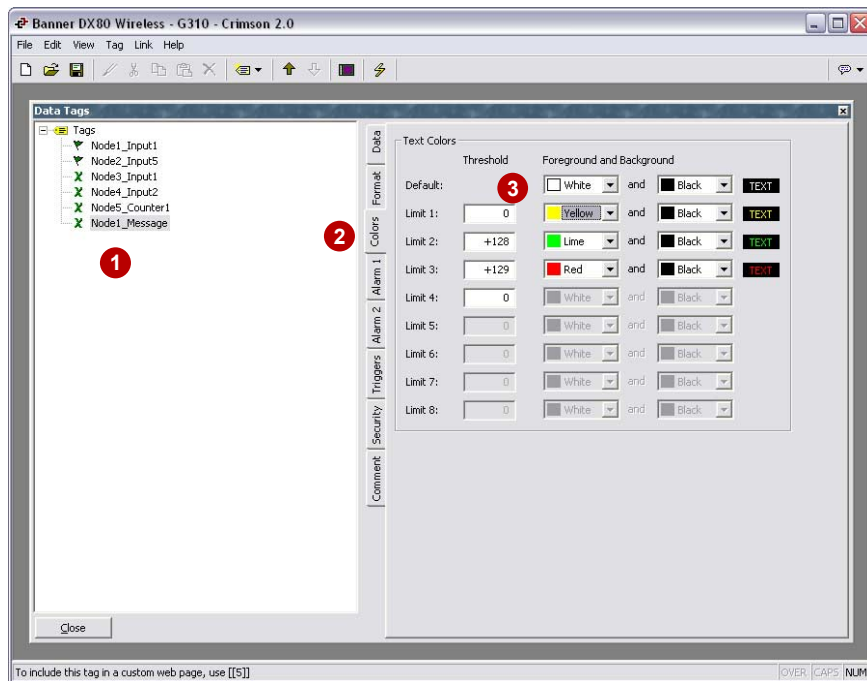


Figure 21

Then, enter the User Interface and insert a Fixed Text object on the display.

1. Type the text Node 1
2. Enter the text object properties (Right-click the object, properties)
3. Change the Background property to Tag Text
4. Select the tag Node1_Message.

The background color of this object will now follow the foreground color of the Node1_Message tag.

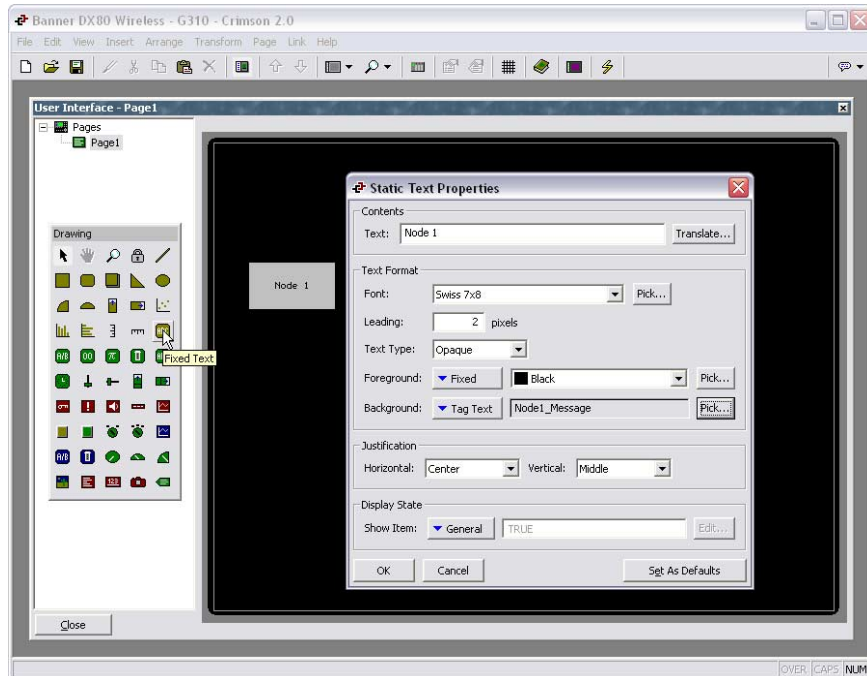


Figure 22

PERFORMING A SITE SURVEY

Banner Specifications And Modbus data

Conducting a Site Survey analyzes the signal strength of the radio communications link between the Gateway and any Node within the network by reporting the number of missed packets, and the relative strength of received data packets. The Site Survey mode can be activated from the Gateway Modbus holding register 15. Only the Gateway can initiate a Site Survey.

Using a Modbus write holding register command, send a control code of 32 (0x20) and the Node number 1-56 (0x01 to 0x38) to the Gateway Modbus holding register for I/O 15.

Gateway Modbus Holding Register																
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Gateway Reg 15	Control Code								Data Field							

Example Command																
Gateway Reg 15	32								02							

When Site Survey is running, the accumulated results are stored in the Gateway's I/O 7 and I/O 8 holding registers. These Modbus registers contain the results of the last 100 samples. The totals are a running tally of the last 100 samples and are continuously updated. Four categories are displayed:

- M = Missed packet
- R = Red – poor signal strength
- Y = Yellow – good signal strength
- G = Green – excellent signal strength

To disable Site Survey, send a control code of 32 (0x20) and a Node number of 0 (0x0).

Gateway Modbus Holding Register																
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Gateway Reg 7	Red Total								Missed Total							
Gateway Reg 8	Green Total								Yellow Total							

Example Results																
Gateway Reg 7	10								0							
Gateway Reg 8	80								10							

Programming Crimson for a site survey

In this example, the site survey will be initiated for Node 3 and the result displayed in the form of bar graphs.

In Crimson, create an integer tag for the Gateway register 15 (Figure 23). By writing in this register, Crimson will send the data to the Gateway.

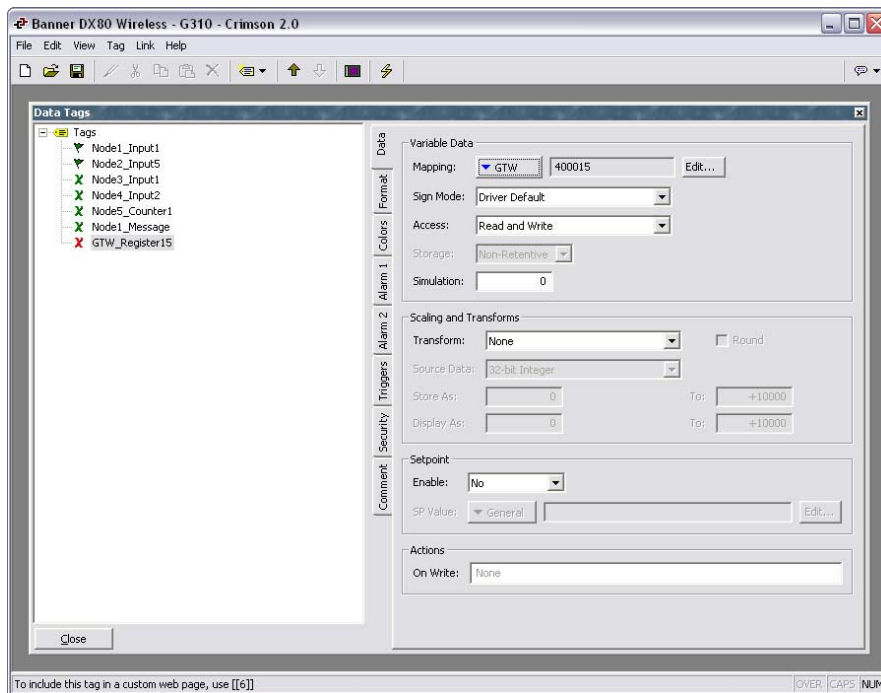


Figure 23

Then create four integer **Formula** tags via the menu >Tag > Create > Formula > Integer (Figure 24). These formulas will contain the survey results.

Rename and change each Tag Value field on the formula as per the table below:

NAME	TAG VALUE FIELD
Missed	[GTW.400007] & 0xFF
Green	[GTW.400008] >> 8
Yellow	[GTW.400008] & 0xFF
Red	[GTW.400007] >> 8

The first operation [GTW.400007] & 0xFF masks the register first 8 bits, so only the value in bit 0 through 7 is left in the formula tag.

The second operation [GTW.400007] >> 8 shifts the whole register 8 bits to the right. Bit 8 through 15 ends up in position 0 through 7, so only the original value in bit 8 through 15 is now left in the tag.

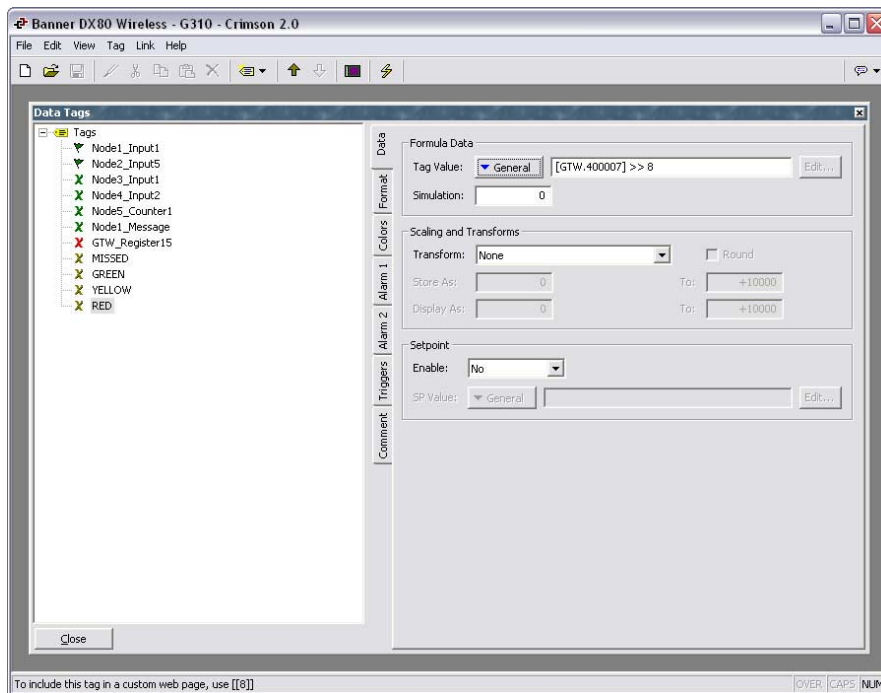


Figure 24

Change the format Minimum and Maximum value in each formula to 0 and 100 respectively (Figure 25).

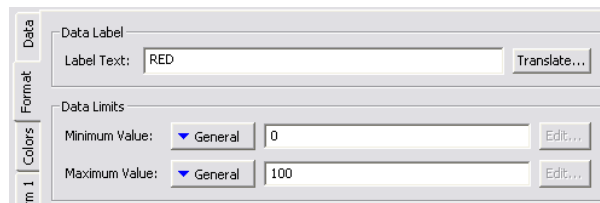


Figure 25

Close the Data Tags window and enter the User Interface window. First, buttons will be created to start and stop a site survey on Node 1.

1. Create a General Button (Figure 26)
2. Enter the button properties and rename it to “Survey Node 1”
3. Click on the Action tab (Figure 27)
4. Change Operation to Change Integer Value
5. Select the Gateway Register 15 tag in the Write To field
6. Enter 0x2001 in data. Here we write in hexadecimal the code 0x20 in byte 2 and the node number 0x01 in byte 1, as per Banner specifications
7. Click OK.

This button is now programmed to send a site survey request for node 1.

Follow the same process to create a second button named “Stop Survey” and containing 0x2000 in data. This button will stop the survey when pressed.

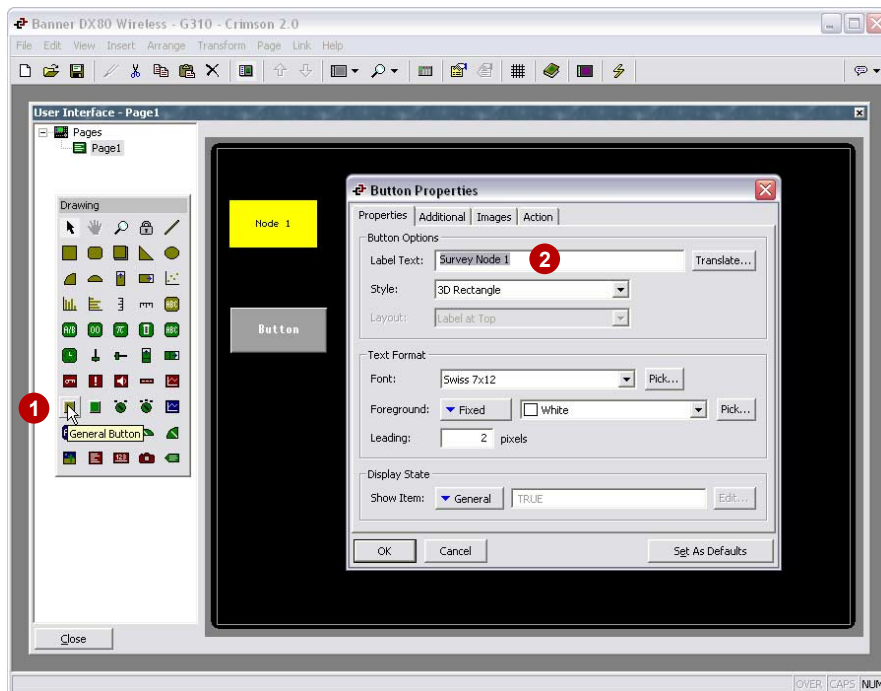


Figure 26

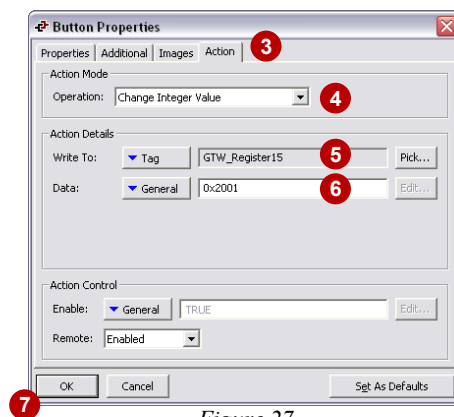


Figure 27

Bar graphs can now be created for each result.

1. Create a rich vertical bar graph (Figure 28)
2. Insert it on the display
3. Select the formula tag MISSED for the Data Source Value
4. Click OK.

Repeat the process three more times and change the Data Source Value to the other result formula tags. The bar graph color can be changed in the bar graph properties on the Format tab.

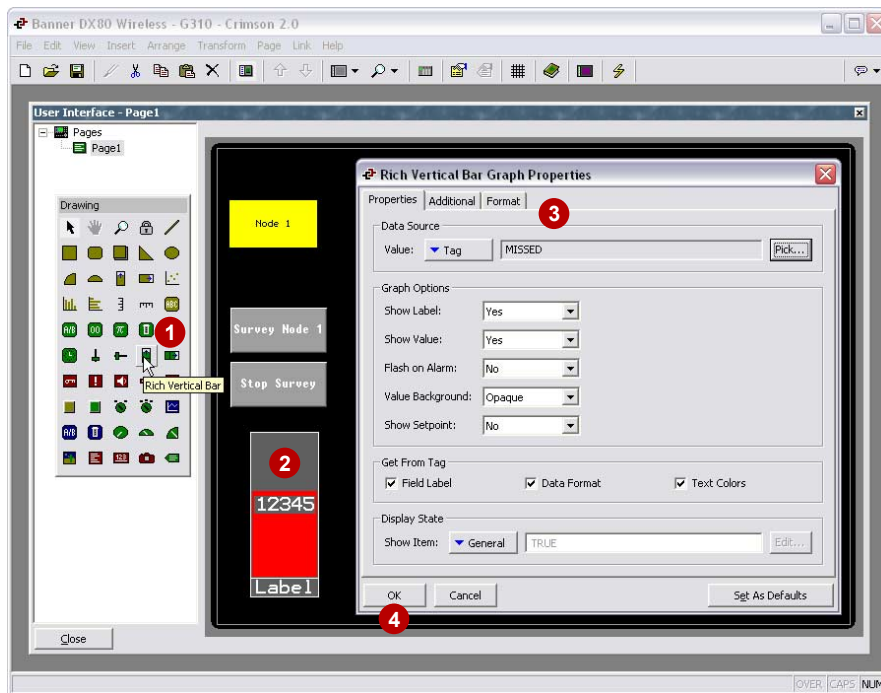


Figure 28

PROTOCOL CONVERSION

Using the G3 embedded protocol converter, Gateway data can be transferred back and forth to any communication able devices connected to the HMI serially, or via Ethernet.

In this example, the G3 will transfer some Input values of Node 1 and 2 to an Allen Bradley SLC on Ethernet.

Enable the G3 Ethernet port and select Protocol 1. Add the protocol Allen Bradley DF1 Master, and then select the device called PLC1.

1. Click the button Add Gateway Block, a new block is created under PLC1 (Figure 29)
2. Select the block, (Figure 30)
3. Change the Start Address to N7:000
4. Change the Block Size to 12

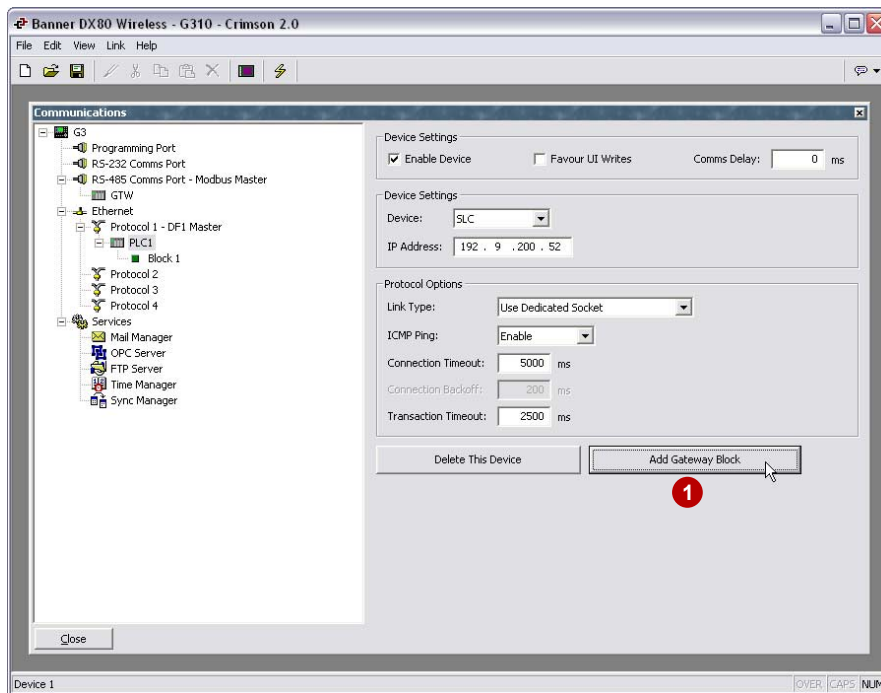


Figure 29

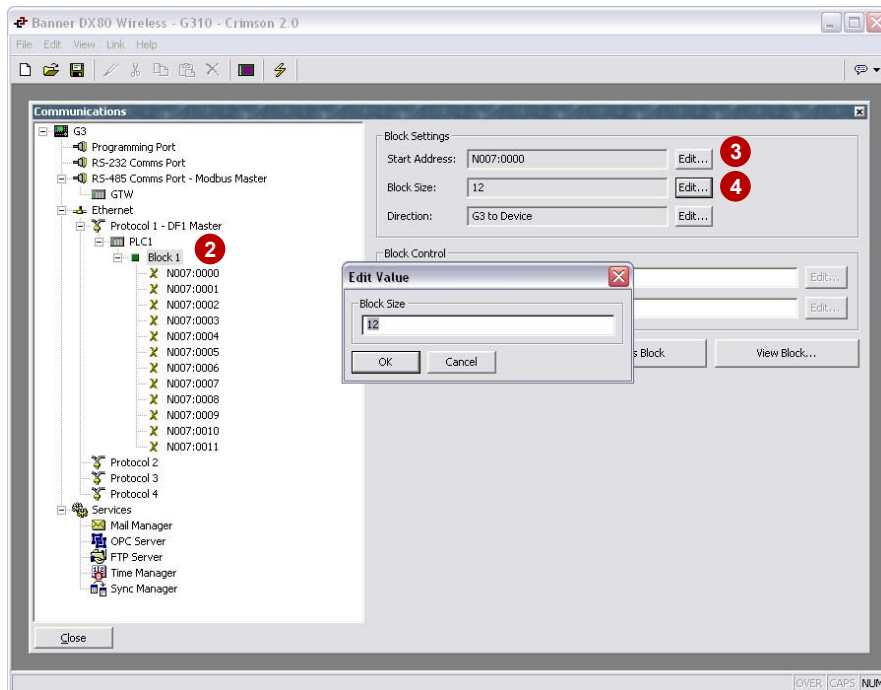


Figure 30

5. Select the first address N7:000. Other devices and tags are now visible (Figure 31)
6. Extend the GTW Device and drag and drop the Holding Register on the first block address

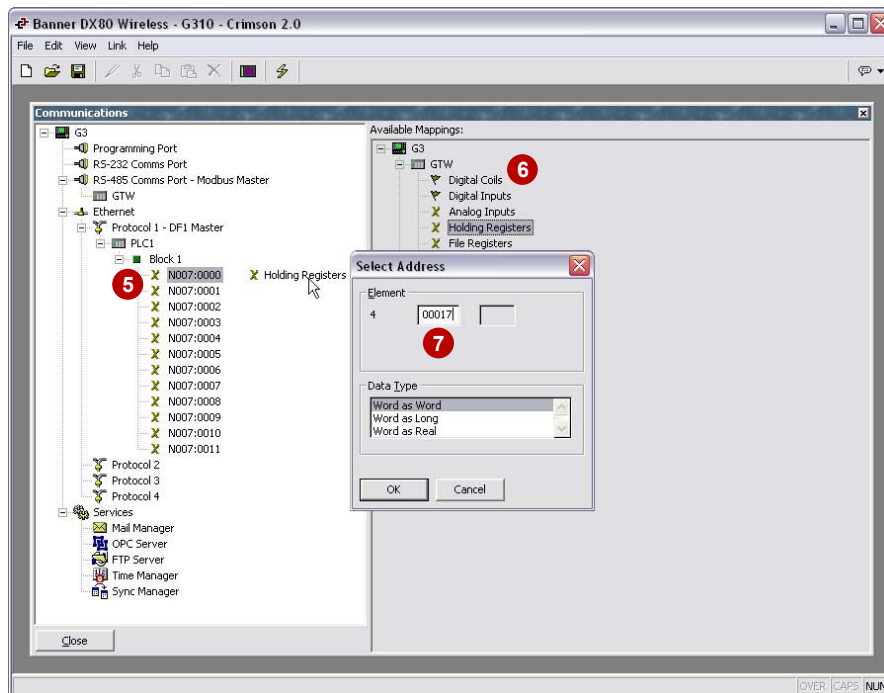


Figure 31

Here, we want to transfer only input registers, so addresses 400017 to 400022 for Node 1 and 400033 to 400038 for Node 2

7. Enter the Modbus address to transfer for Node 1, in this example 400017, and click ok
8. Enter the number of register to exchange, in this example 6
9. Drag and drop again the GTW holding registers to the first available block register
10. Enter the Modbus address to transfer for Node 2, in this example 400022, and click ok
11. Enter the number of register to exchange, in this example 6.

The block is now complete with all inputs from Node 1 and 2.

In order for the PLC to send data to the GTW using the G3 protocol converter, another block with different addresses is required. The block direction can then be reversed to send data from the PLC to the Gateway (the Direction property is available when the block is selected, next to the Start Address and size property).

DATA LOGGING

The G3 embedded data logger is used to record any information coming from serial or Ethernet devices, or internal data tags. In this example, the data logger will be set up to record previously created data tags every five minutes.

Open the Data Logger window from Crimson main menu.

1. Click on the Create Data Log button and select the newly created Log1 (Figure 32)
2. Change the Update Rate to 300 seconds
3. Change the Each File Holds property to 144, so each file will be 12 hours long ($144 \times 5 / 60$)
4. Change the Retain At Most property to 14 to obtain one week of data logging
5. Double click the tags to record.

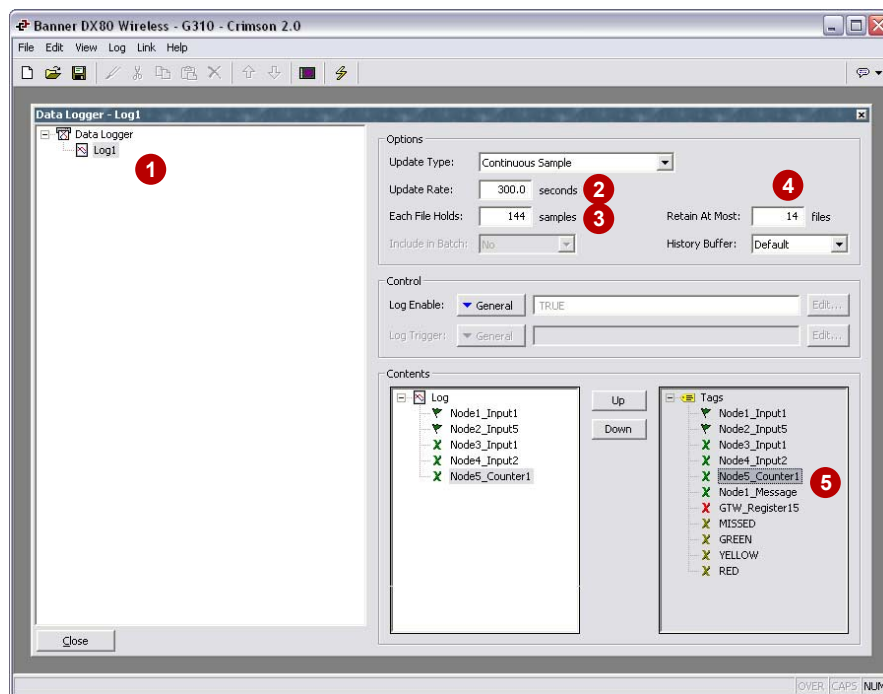


Figure 32