

435NBX – Quick Start Guide

Revision 2

Trademarks

CompactLogix, ControlLogix, & PLC-5 are registered trademarks of Rockwell Automation, Inc. EtherNet/IP is a trademark of the ODVA. MicroLogix, RSLogix 500, and SLC are trademarks of Rockwell Automation, Inc. Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation. All other trademarks and registered trademarks are the property of their holders.

Limited Warranty

Real Time Automation, Inc. warrants that this product is free from defects and functions properly.

EXCEPT AS SPECIFICALLY SET FORTH ABOVE, REAL TIME AUTOMATION, INC. DISCLAIMS ALL OTHER WARRANTIES, BOTH EXPRESSED AND IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR APPLICATION. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular application, Real Time Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams. Except as specifically set forth above, Real Time Automation and its distributors and dealers will in no event be liable for any damages whatsoever, either direct or indirect, including but not limited to loss of business profits, income, or use of data. Some states do not allow exclusion or limitation of incidental or consequential damages; therefore, the limitations set forth in this agreement may not apply to you.

No patent liability is assumed by Real Time Automation with respect to use of information, circuits, equipment, or software described in this manual.

Government End-Users

If this software is acquired by or on behalf of a unit or agency of the United States Government, this provision applies: The software (a) was developed at private expense, is existing computer software, and was not developed with government funds; (b) is a trade secret of Real Time Automation, Inc. for all purposes of the Freedom of Information Act; (c) is "restricted computer software" submitted with restricted rights in accordance with subparagraphs (a) through (d) of the Commercial "Computer Software-Restricted Rights" clause at 52.227-19 and its successors; (d) in all respects is proprietary data belonging solely to Real Time Automation, Inc.; (e) is unpublished and all rights are reserved under copyright laws of the United States. For units of the Department of Defense (DoD), this software is licensed only with "Restricted Rights": as that term is defined in the DoD Supplement of the Federal Acquisition Regulation 52.227-7013 (c) (1) (ii), rights in Technical Data and Computer Software and its successors, and: Use, duplication, or disclosures is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. If this software was acquired under GSA schedule, the U.S. Government has agreed to refrain from changing or removing any insignia or lettering from the Software or documentation that is provided or from producing copies of the manual or media.

© 2015 Real Time Automation, Inc. All rights reserved.

Overview	4
ASCII / CompactLogix Example	5
Description.....	5
Setting up Communications for a CompactLogix PLC.....	5
Setting up Serial Communication	6
Setting up Port 0 for Communication.....	7
ASCII / CompactLogix Example Diagnostics Page	8
ASCII to PLC Direction	9
ASCII to PLC Direction.....	10
PLC to ASCII Direction	11

Overview

This document will explain a sample setup for a 435NBX gateway to communicate to a CompactLogix processor.

This document does not explain the basic ladder logic setup. For that, please refer to the 435_490NBX_LadderLogix.pdf which can be found on the provided CD.

ASCII / CompactLogix Example

Description

Main Page

Device Configuration:

Device Description:	Setting Up Communication from ASCII to ControlLogix on Port 0		
IP Address:	192.168.0.100	Ethernet Link:	Auto-Negotiate ▼
Subnet:	255.255.255.0	MAC Address:	00:03:F4:08:FA:09
Default Gateway:	0.0.0.0	Build Date:	May 27 2015

Setting up Communications for a CompactLogix PLC

PLC Configuration

PLC Type:	CompactLogix ▼		
IP Address:	192.168.0.10		
Controller Slot:	0	0-49; Integrated Ethernet use 0	
Communication Mode:	Connected (Class 3 Explicit) ▼		

Optional

Inter-Message Delay:	0	0-60000 ms	
Heartbeat Tag/File:	<input type="text"/> INT Tag / N Register		

With the above setup, the 435NBX is set up to communicate to a CompactLogix PLC with an IP Address of 192.168.0.10.

The Processor is in Slot 0

We are using Connected Messaging to ensure that the data is moving as reliably as possible.

The inter-message delay is set to 0ms, meaning the 435NBX will communicate with the PLC as fast as possible.

There is no heartbeat tag configured.

Setting up Serial Communication

Serial Configuration

Help

Enable Port 0: <input checked="" type="checkbox"/>	Enable Port 1: <input type="checkbox"/>
Mode: RS232	Mode: RS232
Baud Rate: 19200	Baud Rate: 19200
Parity: None	Parity: None
Data Bits: 8	Data Bits: 8
Stop Bits: 1	Stop Bits: 1
Flow Control: None	Flow Control: None
DTR (Pin 4): High (default) RS232	DTR (Pin 4): High (default) RS232
RTS (Pin 7): High (default) RS232	DTR (Pin 4): High (default) RS232

Save Parameters

RS232

RS232

All of the above serial settings must match the ASCII device that the 435NBX is communicating with. If any one of these fields are incorrect proper communication will not be possible.

Setting up Port 0 for Communication

Port 0 ASCII Configuration

Help
Copy From: Port 0

<< 0 1 >>

ASCII to PLC

Enable Communication:

Data Type: STRING

Tag/File Name: A2PLC_port0

PLC to ASCII

Enable Communication:

Data Type: STRING

Tag/File Name: PLC2A_port0

Define End Case

Character Count: 82 1-4096 chars
Timer: 100 0-30000 ms

Delimiters

Start: 1 [STX] 2 0x02 [NUL] 0 0x00
End: 2 [CR] 13 0x0d [LF] 10 0x0a

Remove Delimiters from ASCII Message:

Add Delimiters to ASCII Message

Start: 0 [NUL] 0 0x00 [NUL] 0 0x00
End: 2 [CR] 13 0x0d [LF] 10 0x0a

Message Queue

Queue Size: 5 0-20 messages
Queue Full Behavior: Discard New Data

Data Conversion

NULL Character Handling: None

Save Parameters

ASCII to CompactLogix:

With the above setup, the 435NBX will start accepting data as soon as the ASCII device sends the character STX. Once the STX is received, the 435NBX will continue to accept characters until one of the following cases has been met:

1. Receive 82 characters.
2. There has been 100ms of no activity on the network.
3. End Delimiters of CR and LF (in that order) were received by the gateway.
 - i. Notice that the Remove Delimiters checkbox is checked, meaning that the STX, CR, and LF will not be passed to the PLC.

CompactLogix to ASCII:

With the above setup, the 435NBX will receive data only when the Length field of the "PLC2A_port0" tag in the PLC is set to a non-zero value. If a value is entered with more than 82 characters, the 435NBX will only send the first 82 characters to the ASCII device, the remainder will be discarded.

When the 435NBX receives the message, it will concatenate the CR and LF to the end of the message because we have defined 2 end delimiters of <CR> and <LF>. Once those characters are added, the message will be transmitted to the ASCII device.

ASCII / CompactLogix Example Diagnostics Page

Once the PLC, Serial, and ASCII parameters have been set, the gateway will start transmitting data between the CompactLogix PLC and the ASCII device.

To view the diagnostics page, click the **Diagnostics** button, navigate to Port 0, and select **ASCII to PLC** direction.



Figure 1

The above screen shot shows a connected status to the CompactLogix PLC. It took 1 attempt to connect to make the connection. If the number of Connection Attempts is incrementing there is an issue with the communication between the PLC and the 435NBX. This could be due to a timeout or an error.

ASCII to PLC Direction

Port 0 Diagnostics << 0 1 >> Help

ASCII to CompactLogix ▾

Last message sent to CompactLogix (12 chars)
0000: 48 65 6C 6C 6F 20 57 6F 72 6C 64 21 Hello World!
Ref 1 →

Next message stored in ASCII queue (8 chars - 2/5 Messages Queued)
0000: 47 6F 6F 64 62 79 65 2E Goodbye.
Ref 3 →

Current message being processed (59 chars) ASCII message end case not yet met
0000: 57 61 69 74 69 6E 67 20 66 6F 72 20 65 6E 64 20 Waiting for end
0016: 64 65 6C 69 6D 69 74 65 72 73 2C 20 74 69 6D 65 delimiters, time
0032: 20 6F 75 74 2C 20 6F 72 20 4D 61 78 20 43 68 61 out, or Max Cha
0048: 72 61 63 74 65 72 73 2E 2E 2E 2E racters....

Clear Buffers

Send Test Message to CompactLogix

Send Test Message

Diagnostic Counters

ASCII Event **Ref 2** →
Delimiter: 3 Length: 0 Timeout: 0 Discards: 0

Read Handshake Message from CompactLogix
OK: 5178 Error: 0 Last Error:

Write ASCII Message to CompactLogix
OK: 1 Error: 0 Last Error:

Figure 2

Figure 2 shows the first message sent to the PLC, Ref 1, of “Hello World!” It was defined by end delimiters because the ASCII Event Delimiter Counter is incrementing, Ref 2. Each time a message is processed, its end state is recorded.

In the above example, 3 messages meeting the defined end delimiters, CR and LF, have been received. The third complete message is not displayed because it is in the Queue buffer, Ref 3. This is identified in the gateway by “2 Messages Queued” next to the *Next message stored in ASCII queue* buffer. Only the next message to be sent to the PLC is visible in the queue.

ASCII to PLC Direction

Port 0 Diagnostics << 0 1 >> Help

ASCII to CompactLogix ▾

Last message sent to CompactLogix (12 chars)

```
0000: 48 65 6C 6C 6F 20 57 6F 72 6C 64 21      Hello World!
```

Next message stored in ASCII queue (8 chars - 2/5 Messages Queued)

```
0000: 47 6F 6F 64 62 79 65 2E      Goodbye.
```

Current message being processed (59 chars) **ASCII message end case not yet met**

```
0000: 57 61 69 74 69 6E 67 20 66 6F 72 20 65 6E 64 20      Waiting for end
0016: 64 65 6C 69 6D 69 74 65 72 73 2C 20 74 69 6D 65      delimiters, time
0032: 20 6F 75 74 2C 20 6F 72 20 4D 61 78 20 43 68 61      out, or Max Cha
0048: 72 61 63 74 65 72 73 2E 2E 2E 2E      racters....
```

Clear Buffers

Send Test Message to CompactLogix

Send Test Message

Diagnostic Counters

ASCII Event	Ref 1	Length: 0	Timeout: 0	Discards: 0
Delimiter: 3				
Read Handshake Message from CompactLogix		Error: 0	Last Error:	
OK: 5178				
Write ASCII Message to CompactLogix	Ref 2	Error: 0	Last Error:	Ref 3
OK: 1				

The second message “GoodBye.” will be held until the length field of the PLC string is set to 0. The non-zero length will also trigger the Read Handshake Message counter to increment, **Ref 1**.

The data in the *Current message being processed* buffer has not reached an end case. It will not be considered complete until the gateway receives a [CR][LF], 100ms pass after receiving a character, or the length reaches 82 characters.

The *Write ASCII Message to PLC OK* counter indicates the number of messages sent to the PLC, **Ref 2**. This value should be equal to ASCII Event Delimiter Count + Length Count + Timeout Count. If these values are not equal, then the length field in the PLC is non-zero.

If the *ASCII Event Discard* counter is incrementing, **Ref 3**, the gateway is receiving messages faster than the PLC is processing them.

PLC to ASCII Direction

To view the diagnostics page, click the **Diagnostics** button, navigate to Port 0, and select **PLC to ASCII** direction.

Port 0 Diagnostics << 0 1 >> Help

TCP/IP Status: Connected **Connection Attempts:** 1

CompactLogix to ASCII ▾

Last message received from CompactLogix (42 chars)

```
0000: 53 65 6E 64 69 6E 67 20 44 61 74 61 20 46 72 6F   Sending Data Fro
0016: 6D 20 41 20 50 4C 43 20 54 6F 20 41 6E 20 41 53   m A PLC To An AS
0032: 43 49 49 20 44 65 76 69 63 65                   CII Device
```

Send Test Message to ASCII

Send Test Message

Diagnostic Counters

Read ASCII Message from CompactLogix
OK: 28229 Error: 0 Last Error:

Write Handshake Message to CompactLogix
OK: 2 Error: 0 Last Error:

Clear Counters

For moving data from the PLC to the ASCII device, the length field of the tag in the PLC needs to be set to non-zero. The gateway monitors the length field for a non-zero value. Once a non-zero value is seen, the gateway will read the data and send it to the ASCII device. It will also clear the tag and reset the length field to 0. This handshake lets the PLC know the data has been processed.

In the above example “Sending Data From A PLC to An ASCII Device” was written to a tag, changing the length field to 42. The non-zero length of 42 triggered the gateway to process the message.

The *Read ASCII Message from PLC OK* counter should always be incrementing. This is a heartbeat counter that increments every time the gateway reads the PLC tag length to determine if there is a new message, **Ref 1**.

The *Write Handshake Message to PLC OK* counter will increment when a message is received, **Ref 2**.