

CONTREX, Inc

DeviceNet Startup Guide

December 22, 2003

Overview:

This guide offers a short-form startup procedure and additional comments regarding the CONTREX CX1010 controller connections to a DeviceNet network. The CX1010 manual covers the setup parameters in chapter 5, under the Main Menu\Device Configure\DeviceNet Setup. Detailed DeviceNet specifications are detailed in a separate CX-Series DeviceNet Card manual. This short-form guide contains information from both sources, and simplifies the procedure for most users who meet the "Assumptions" criteria noted below.

Assumptions:

The user is generally familiar with DeviceNet terminology, and wiring practices.
The user is familiar with the specific Host/Scanner card that will control this network.
The Scanner card is probably mounted in a PLC (vs. PC) host controller.
The user is familiar with writing PLC code to manipulate numbers. In particular there may be a need to move decimal points around by multiplying, or dividing, by 10's.
The primary communications mode will be "Polling", rather than "Explicit Messaging".
The data list (CX1010 parameters of interest) includes no more than 4 "write", nor more than 4 "read" parameters.

Setup the CX1010:

Go through the Menu and find Main Menu\Device Configure\DeviceNet Setup.
Enter a MAC ID value between 1 and 63. This will be the Node Address for this controller on your network. Be sure you do not have 2 devices with the same address.

Enter a baud rate. Choices are 125, 250, or 500 kbaud. All devices on the network must communicate with the Scanner Card at the same network speed.

Enter the unique product serial number for this network card. It would be found with the information that came with the card. If you cannot locate it, you may proceed with the setup anyway, as most host systems probably do not use this information anyway.

Proceed to page 2 of the CX1010 DeviceNet Setup menu. Enter the list of parameters that will be communicated over the network. The correct Parameter Number may be found from the lists in the CX1010 manual. Alternately, you could find the parameter

through the front panel of the CX1010 itself. First, find the target parameter via the Menus. Then obtain the parameter number off the Help screen for that parameter. Typical parameters might include Setpoint (101) and Scaled Feedback (40).

The selected parameter numbers must be entered from the top of the write, or read, list. If selecting less than 4 items, enter a "0" as the parameter number for the remaining slots on the list. Note that a Control Parameter (CP-XX) can be written to, but it also can be on the read list if you wish to have the host monitor, or verify, what value is actually present in that parameter.

Note regarding Allen Bradley's DeviceNet Manager:

The DeviceNet option card has now been configured for polling through the above keyboard entry procedure. It is NOT necessary to use Allen Bradley's DeviceNet Manager software, although an experienced user could have configured the polling connection (Assembly Queue) through that software if desired. If you do use the Manager, verify that the desired polling parameters do appear on the appropriate CX1010 menu screens. If necessary, try sending a "Start Service" command from the Manager after configuration. There is a CONTREX option card EDS file available for use with that software.

Setup the Scanner (Host):

Refer to the instructions for the particular model of scanner for detail information. It will typically ask for the number of bytes produced, or consumed by the CX1010.

Consumed bytes are those the host will send to the CX1010, and contain the data that you wish to "write" to the CX1010 parameters.

Produced bytes are those the host expects to receive back from the CX1010, and contain the data from the "read" list parameters.

Each parameter on the list requires 6 bytes. If all 4 slots were utilized for both the write and read lists, then both the consumed and the produced byte count would be $4 \times 6 = 24$ bytes.

Data Format & Definition:

Each parameter value is transmitted over the network as 6 bytes.

The first 4 bytes (2 words) contain a signed long integer.

The next byte, the 5th byte, is a resolution byte. It represents the number of digits after the decimal point. Here are some examples:

<u>Real value</u>	<u>Signed integer</u>	<u>Resolution byte</u>
1010	1010	0
12.34	1234	2
1.50	150	2
-200	-200	0

The 6th byte is the Definition byte. It is NOT important. Treat it as a placeholder only. It can be used for obscure purposes in other types of serial communications, but it was included in the DeviceNet protocol only to create an even number of bytes. PLC's often assign variable names, and memory addresses, by the word (2 bytes) It is useful to keep CX1010 parameters in "even byte" groups.

Shortcut tip:

Most users do NOT need to dynamically monitor decimal location. Since you are writing and reading carefully selected and known parameters, you already know where the decimal points are. If a setpoint ratio is always near 1.000, then you know the format will be x.xxx. You would permanently set the resolution byte for writing a setpoint ratio to the value 3. Then, to send a ratio of 1.123, assign the 1st 4 bytes a value (long integer) of 1123, and append a resolution byte of 3, and a definition byte of zero, for a total of 6 bytes. If you are reading a setpoint ratio, you can ignore the resolution byte (you already know it will be 3), and ignore the definition byte (not important). You would receive a long integer (1st 4 bytes) of 1123, and you would routinely divide it by 1000 to place the decimal correctly. The rest of your PLC code would then accept this as a real value of 1.123.

Since you will be ignoring incoming resolution and definition bytes, you only need to act upon the 1st 4 bytes (2 words) of any parameter in the read list.

Since the resolution byte can be predetermined (decimal places), and the definition byte is not important, you can easily pre-establish the 2 bytes to append to the outgoing "write" parameters. If you treat these as 1 word, or as a 2 byte integer, the following values will place the correct resolution byte (decimal places) in the "high" byte of this word or integer:

<u>decimal places</u>	<u>value of 2 byte integer</u>
0	0
1	256
2	512
3	768
4	1024

Explicit Messaging:

While the CX1010 DeviceNet option card is capable of explicit messaging, most users do not use this feature. There are some initializing activity that requires some explicit messages, but the scanner card (host) will take care of these automatically. The User does not usually have to deal with these at all. Just follow the setup procedures for your scanner.

An explicit message can be sent to change the communications time-out values in the CX1010 card. This is known as the EPR, or Expected Packet Rate. The default time-out is 10 sec, but can be set to indefinite. However, the easiest solution is to tell the scanner to "Ping" the CX1010. The scanner will send a "no-op" message at regular intervals to keep the node alive if scanning is interrupted for any reason.

Faults and Errors:

An explicit message can query the CX1010 to determine a communication fault, or reason for a rejected parameter value. Most users do NOT go to that much trouble. The scanner itself will report many types of communications errors in an easier to use format. Most users that need to verify that a "write" action was successful will use a write & read sequence. Just add the write parameter to the read list. Then write the new value, wait 1 scan, and compare the read value to the original write value.

Auto/Manual Selection:

The rejection of a write value is not always an error. This feature can be used to toggle between an Auto Mode (PLC determines setpoint), and a Manual Mode (CX1010 operator manually enters a setpoint). Of course the operator can use other means to acquire control over the setpoint (i.e. direct the PLC scan to a block parameter and change blocks via a switch), but there also is a way for the PLC to make that decision and override the operator's setpoint if needed. The PLC would send a "reasonable" setpoint value any time it wanted to take control. Since polling happens very rapidly, any setpoint entered by the operator would be immediately over-written by the value from the PLC.

When the PLC wishes to relinquish control to the operator, it simply sends an "unreasonable" setpoint value to the CX1010. Usually you choose a very large number, (all 9's). This number must be in excess of the allowable max setpoint parameter. The CX1010 rejects that setpoint, and retains the present or existing setpoint (presumably the value entered by the operator). This technique addresses the fact that the polling list, or polling action, is not easily modified on-the-fly. It is easier to just substitute the "unreasonable" number for the polled parameter value. The PLC can also use this technique to monitor the operator's manually entered setpoints. By reading the operator's entry, the PLC can over-write it with a different number if it is determined that the operator entered an inappropriate value.

Change from CX1000 (Ext Power Supply Requirement):

The CX1010 implements DeviceNet with the same software routines as did the CX1000. However, there is a Hardware difference. The DeviceNet option card requires more power (the spec says 250ma at 5vdc, but it actually draws less) than is available from the controller. This power requirement (applied at JB2) is for the on-board processor, and should not be confused with the normal external 24vdc power that runs the DeviceNet network itself (applied at JB1). We have always recommended an external power supply at 5vdc, 300 ma. However, the CX1000 had a 5vdc accessory supply rated at 200ma that was often used in spite of our recommendation. The CX1010 does NOT have a 5vdc accessory supply, but now has a 12vdc supply that is intended for encoder use. This 12vdc is NOT acceptable for the DeviceNet card. An external supply is required. You may use any brand or style, but for reference, you could use DigiKey catalog number T309-ND, rated 300ma at 5vdc +/- 5% at approx \$10 cost.