INDUSTRIAL CONTROL COMMUNICATIONS, INC.

Metasys N2 Slave Driver Manual

February 17, 2020

TABLE OF CONTENTS

1 Metasys N2 Slave

1.1 Overview

This driver supports the Johnson Controls Metasys® N2 Open protocol as a network slave. Some notes of interest are:

- Supports fully-configurable analog input, analog output, binary input, binary output, internal float, internal integer, and internal byte object types.
- The Metasys device type for the driver is VND.

1.2 Connections

This section describes the typical connections used for a Millennium Series gateway.

Connect the N2 bus wiring to the RS-485 port by using twisted-pair cable connected as shown in Figure 1. Connect the N2+ wire to terminal “A”, the N2- wire to terminal “B”, and the network ground wire to terminal “GND”. Also install jumper wires connecting terminal “A” to terminal “Y”, and terminal “B” to terminal “Z”. Continue this connection scheme throughout the remainder of the network. Always connect each unit in a daisy-chain fashion, without drop lines, star configurations, etc. For further N2 network wiring requirements and procedures, please refer to the appropriate JCI network installation documentation.

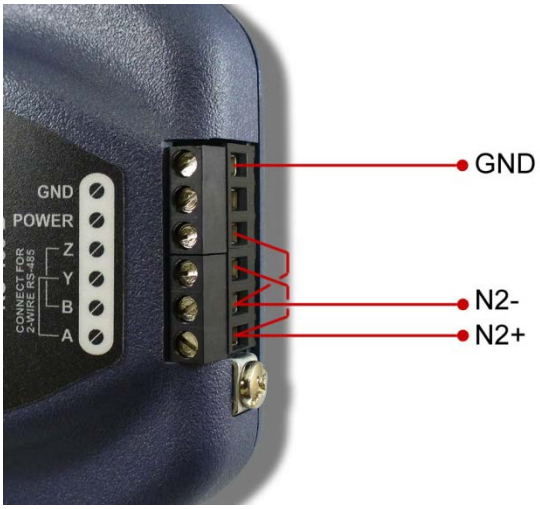

Figure 1: N2 Bus Cable Connection to RS-485 Port

1.3 Slave Settings

Baud Rate

Fixed at 9600 baud.

Parity

Fixed at No Parity (1 Stop Bit).

Timeout Time

Defines the maximum number of milliseconds for a break in network communications before a timeout event will be triggered. To disable timeout processing, set this field to 0.

Response Delay

Defines the time (in milliseconds) that the driver waits before responding to master requests. This is a useful feature for certain master devices or infrastructure components (such as radio modems) that may require a given amount of time to place themselves into a "receiving mode" where they are capable of listening for slave responses. If no delay is required, setting this field to 0 instructs the driver to send its responses as soon as possible.

1.4 Node Settings

Address

Defines the station address (1…255) for this node.

1.5 Metasys Object Settings

The N2 slave driver hosts Metasys objects which contain a variety of different attributes for an N2 master to access. The Metasys specification allows a maximum of 256 instances of each object type.

- **Analog input** (AI) objects are used for monitoring analog status items. AI objects support low alarm limits, low warning limits, high warning limits, high alarm limits and differential values. Change of state (COS), alarm and warning functions can also be enabled. An AI object will accept an override command, but will not change its actual value or indicate override active. A "multiplier" is associated with the object, which can provide scaling between the network value and raw (database) value.
- **Analog output** (AO) objects are used for setting and monitoring analog control and configuration items. An AO value can be modified by issuing an override command. Issuing a release command will not cause the AO to automatically return to its pre-override value, nor will the AO automatically return to its pre-override value after a certain time period of no communication. A "multiplier" is associated with the object, which can provide scaling between the network value and raw (database) value.
- **Binary input** (BI) objects are used for monitoring discrete (digital) status items. BI objects support COS, alarm enabling and normal/alarm status indications. A BI object will accept an override command, but will not change its actual value or indicate override active. A "bitmask" is associated with the object, which provides mapping information between the network value and specific bits in the database.
- **Binary output** (BO) points are used for setting and monitoring discrete control and configuration items. A BO value can be modified by issuing an override command. Issuing a release command will not cause the BO to automatically return to its pre-override value, nor will the BO return to its pre-override value after a certain time period of no

communication. A “bitmask” is associated with the object, which provides mapping information between the network value and specific bits in the database.

- **Internal float** (ADF) objects are used for setting and monitoring internal floating point values. An internal float value can be modified by issuing an override command or a write attribute command. Issuing a release command will not cause the internal float to automatically return to its pre-override value, nor will the internal float automatically return to its pre-override value after a certain time period of no communication. A “multiplier” is associated with the object, which can provide scaling between the network value and raw (database) value.
- **Internal integer** (ADI) objects are used for setting and monitoring internal integer values. An internal integer value can be modified by issuing an override command or a write attribute command. Issuing a release command will not cause the internal integer to automatically return to its pre-override value, nor will the internal integer automatically return to its pre-override value after a certain time period of no communication. A “multiplier” is associated with the object, which can provide scaling between the network value and raw (database) value.
- **Internal byte** (BD) objects are used for setting and monitoring internal byte values. An internal byte value can be modified by issuing an override command or a write attribute command. Issuing a release command will not cause the internal byte to automatically return to its pre-override value, nor will the internal byte automatically return to its pre-override value after a certain time period of no communication. A “multiplier” is associated with the object, which can provide scaling between the network value and raw (database) value.

1.5.1 Analog Input Object Settings

Object Name

This field is a description of the Metasys object. It is not used other than to serve as a reference for the user. Enter a string of between 1 and 32 characters in length.

Instance

Defines the Metasys object’s instance number (1…256).

Database Address

Defines the database address where the Metasys object’s value will reside. The configuration studio will not allow entry of a database address that will cause the object’s value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

Data Type

Specifies how the object’s value will be stored in the database. This defines how many bytes will be allocated, whether the value should be treated as signed or unsigned, and whether the value should be interpreted as an integer or a floating point number. Select the desired data type from this dropdown menu.

Multiplier

The amount that associated network values are scaled by prior to being stored into the database or after being retrieved from the database. Upon retrieval from the database, raw data is multiplied by the multiplier to produce a network value (to send to the master.)

1.5.2 Analog Output Object Settings

Object Name

This field is a description of the Metasys object. It is not used other than to serve as a reference for the user. Enter a string of between 1 and 32 characters in length.

Instance

Defines the Metasys object's instance number (1…256).

Database Address

Defines the database address where the Metasys object's value will reside. The configuration studio will not allow entry of a database address that will cause the object's value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

Data Type

Specifies how the object's value will be stored in the database. This defines how many bytes will be allocated, whether the value should be treated as signed or unsigned, and whether the value should be interpreted as an integer or a floating point number. Select the desired data type from this dropdown menu.

Multiplier

The amount that associated network values are scaled by prior to being stored into the database or after being retrieved from the database. Upon retrieval from the database, raw data is multiplied by the multiplier to produce a network value (to send to the master.) Similarly, network values (received from the master) are divided by the multiplier before being stored into the database.

1.5.3 Binary Input Object Settings

Object Name

This field is a description of the Metasys object. It is not used other than to serve as a reference for the user. Enter a string of between 1 and 32 characters in length.

Instance

Defines the Metasys object's instance number (1…256).

Database Address

Defines the database address where the Metasys object's value will reside.

Data Type

Fixed at 8-Bit Unsigned.

Bitmask

Specifies which bit(s) in the byte designated by the "Database Address" that the binary object will map to. This mechanism allows up to 8 binary objects to be simultaneously assigned to one database address (each binary object mapping to a single bit of that byte in the database). It is possible to map binary objects to multiple bits within the designated database location.

The effect of the "Bitmask" field when reading: When the current state of a binary object is read by a Metasys master, the bitmask is used to determine the state of the object by inspecting the value in the designated database address at the bit location(s) indicated in the bitmask. If all of the bit locations at the designated database address indicated by a checkmark in the bitmask are set, then the object's state will be returned as "1". Else, the object's state will be returned as "0".

1.5.4 Binary Output Object Settings

Object Name

This field is a description of the Metasys object. It is not used other than to serve as a reference for the user. Enter a string of between 1 and 32 characters in length.

Instance

Defines the Metasys object's instance number (1…256).

Database Address

Defines the database address where the Metasys object's value will reside.

Data Type

Fixed at 8-Bit Unsigned.

Bitmask

Specifies which bit(s) in the byte designated by the "Database Address" that the binary object will map to. This mechanism allows up to 8 binary objects to be simultaneously assigned to one database address (each binary object mapping to a single bit of that byte in the database). It is possible to map binary objects to multiple bits within the designated database location. Such a configuration allows (for example) the modification of multiple selected database bits via a single binary output.

The effect of the "Bitmask" field when reading: When the current state of a binary object is read by a Metasys master, the bitmask is used to determine the state of the object by inspecting the value in the designated database address at the bit location(s) indicated in the bitmask. If all of the bit locations at the designated database address indicated by a checkmark in the bitmask are set, then the object's state will be returned as "1". Else, the object's state will be returned as "0".

The effect of the "Bitmask" field when writing: When the current state of a binary object is overridden to "1" by a Metasys master, then the bit(s) in the designated database address indicated by a checkmark in the bitmask are set. Similarly, when the current state of the object is overridden to "0", then the bit(s) in the designated database address indicated by a checkmark in the bitmask are cleared.

1.5.5 Internal Float Object Settings

Object Name

This field is a description of the Metasys object. It is not used other than to serve as a reference for the user. Enter a string of between 1 and 32 characters in length.

Instance

Defines the Metasys object's instance number (1…256).

Database Address

Defines the database address where the Metasys object's value will reside. The configuration studio will not allow entry of a database address that will cause the object's value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

Data Type

Specifies how the object's value will be stored in the database. This defines how many bytes will be allocated, whether the value should be treated as signed or unsigned, and whether the value should be interpreted as an integer or a floating point number. Select the desired data type from this dropdown menu.

Multiplier

The amount that associated network values are scaled by prior to being stored into the database or after being retrieved from the database. Upon retrieval from the database, raw data is multiplied by the multiplier to produce a network value (to send to the master.) Similarly, network values (received from the master) are divided by the multiplier before being stored into the database.

1.5.6 Internal Integer Object Settings

Object Name

This field is a description of the Metasys object. It is not used other than to serve as a reference for the user. Enter a string of between 1 and 32 characters in length.

Instance

Defines the Metasys object's instance number (1…256).

Database Address

Defines the database address where the Metasys object's value will reside. The configuration studio will not allow entry of a database address that will cause the object's value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

Data Type

Specifies how the object's value will be stored in the database. This defines how many bytes will be allocated, whether the value should be treated as signed or unsigned, and whether the value should be interpreted as an integer or a floating point number. Select the desired data type from this dropdown menu.

Multiplier

The amount that associated network values are scaled by prior to being stored into the database or after being retrieved from the database. Upon retrieval from the database, raw data is multiplied by the multiplier to produce a network value (to send to the master.) Similarly, network values (received from the master) are divided by the multiplier before being stored into the database.

1.5.7 Internal Byte Object Settings

Object Name

This field is a description of the Metasys object. It is not used other than to serve as a reference for the user. Enter a string of between 1 and 32 characters in length.

Instance

Defines the Metasys object's instance number (1…256).

Database Address

Defines the database address where the Metasys object's value will reside. The configuration studio will not allow entry of a database address that will cause the object's value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

Data Type

Specifies how the object's value will be stored in the database. This defines how many bytes will be allocated, whether the value should be treated as signed or unsigned, and whether the value should be interpreted as an integer or a floating point number. Select the desired data type from this dropdown menu.

Multiplier

The amount that associated network values are scaled by prior to being stored into the database or after being retrieved from the database. Upon retrieval from the database, raw data is multiplied by the multiplier to produce a network value (to send to the master.) Similarly, network values (received from the master) are divided by the multiplier before being stored into the database.

1.6 Override Release Event (PicoPort/Mirius Only)

Each commandable object, i.e. Output and Internal objects, can optionally include an override release event. This adds the ability to detect when a release command has been received for the object. The event is assigned to a byte in the database. When a release command is received for the associated object, the event database location is set to a value of 1.

Release Event Database Address

Specifies the database address to use for detecting that a release command has been received for the object.

ICC

INDUSTRIAL CONTROL COMMUNICATIONS, INC.

230 Horizon Drive, Suite 100
Verona, WI USA 53593
Tel: [608] 831-1255

http://www.iccdesigns.com

Printed in U.S.A