AF91DNA Instruction Leaflet

DESCRIPTION

The AF91DNA is a DeviceNet communication module that allows the AF91 drive to be controlled from a DeviceNet network. The AF91DNA is designed to plug into the RJ45 connector on the bottom of the AF91 drive where the remote terminal would be connected. The AF91DNA has a DIN rail connector on the back so that it can be mounted on a DIN rail near the AF91 and comes with a 12 inch cable to connect the communications from the drive to the interface. The AF91DNA allows the user to configure and control the functions of the AF91 from a DeviceNet network. The AF91DNA can communicate on the DeviceNet network at 125K, 250K and 500K. A relay contact is provided on the AF91DNA to connect to the AF91 so that when communications is disrupted between the AF91 and the AF91DNA, the AF91 can be configured to perform a shutdown.

FEATURES

- Configuration of the AF91 drive parameters over DeviceNet
- Control of the AF91 from DeviceNet
  - Start/Stop
  - Direction
  - Fault reset
  - Speed control
  - Acceleration/Deceleration control
- Hand - Auto control

⚠ Warning

Upon power-up, the AF91DNA initialization routine programs the function and polarity of terminal 5 of the AF91 inverter to the function specified by the TB5-Comm Loss attribute in the AF91 Protections Object (Class = 166 Instance = 1 Attribute ID = 14). This new function is not necessarily the same as the Cutler-Hammer out-of-box factory default for terminal 5. If the AF91 is to be returned to a stand alone, non-DeviceNet application, the function of terminal 5 must be reprogrammed to match the requirements of the new application. FAILURE TO DO SO COULD RESULT IN EQUIPMENT DAMAGE, PERSONAL INJURY, OR DEATH.

AF91DNA Description of Operation

The AF91DNA DeviceNet Adapter is used to provide a Polled I/O Connection between the Cutler-Hammer AF91 Drive and a DeviceNet master. After proper installation and configuration of the AF91DNA and AF91 drive, the AF91 can be continuously controlled and monitored over a DeviceNet network. In the default mode of operation, the AF91DNA provides 4 bytes of Polled Response data and consumes 4 bytes of Polled Command data.

The Polled Command message from the DeviceNet master sends the following data:

1. Whether the AF91 is controlled (Start/Stop) by the DeviceNet master or input signals on the terminal block.
2. The speed reference source for the AF91 (DeviceNet or local Terminal Block).
3. Start/Stop/Fault Reset signals to the AF91 (if controlled from DeviceNet).
4. The Speed Reference value (in RPM) for the AF91 (if controlled from DeviceNet).

The Polled Response message from the AF91DNA returns the following data:

1. The state of the AF91 drive, running or stopped.
2. The states of the inputs and outputs on the AF91 terminal block.
3. The actual speed of the AF91 (in RPM).

Through the use of DeviceNet configuration software such as Cutler-Hammer's NetView, the size and type of information contained in the Polled I/O messages can be modified to fit the user's application. Additional Polled Command data that can be sent to the drive includes Acceleration and Deceleration time. Polled Response data from the AF91DNA can include fault status, PID feedback data, control and speed reference point (local or DeviceNet), and the value of the input speed reference, for applications in which the reference is provided locally via the terminal block. Please refer to the AF91DNA Configuration section later in this manual to properly select which Input Assembly Object and Output Assembly Object are used for the Polled I/O Connection. Refer to the Assembly Object section of Appendix A for detailed information on the format of the Polled I/O message data.

The AF91DNA also supports Explicit Messages from the DeviceNet master to the AF91 drive. The DeviceNet master can use explicit messages to read and modify the configuration and operation of the AF91 drive and AF91DNA adapter, as well as perform additional functions such as resetting the drive and/or adapter. Refer to Appendix A at the end of this manual for a complete list of supported explicit message services and DeviceNet objects.

AF91DNA Specifications

ENVIRONMENT

- Ambient temperature:
  Operating: -10 to +55°C
  Storage: -25 to 70°C
- Humidity: 20 to 90% non-condensing
- Maximum Altitude: 1000 meters
- Vibration: 0.6G 10-55 Hz

COMMUNICATIONS

- ODVA 2.0 Compliant
- IO Slave Messaging: Poll and Explicit
- Data rates: 125K, 250K, 500K Baud

ELECTRICAL

- DeviceNet
  Network supply voltage: 11 to 25 volts DC nominal, 30 volts DC max.
  Network input current: 30 ma typical, 70 ma inrush @ 24 volts DC
  Power consumption: 1W max
- Other
  All other power derived from AF91 inverter power supply.

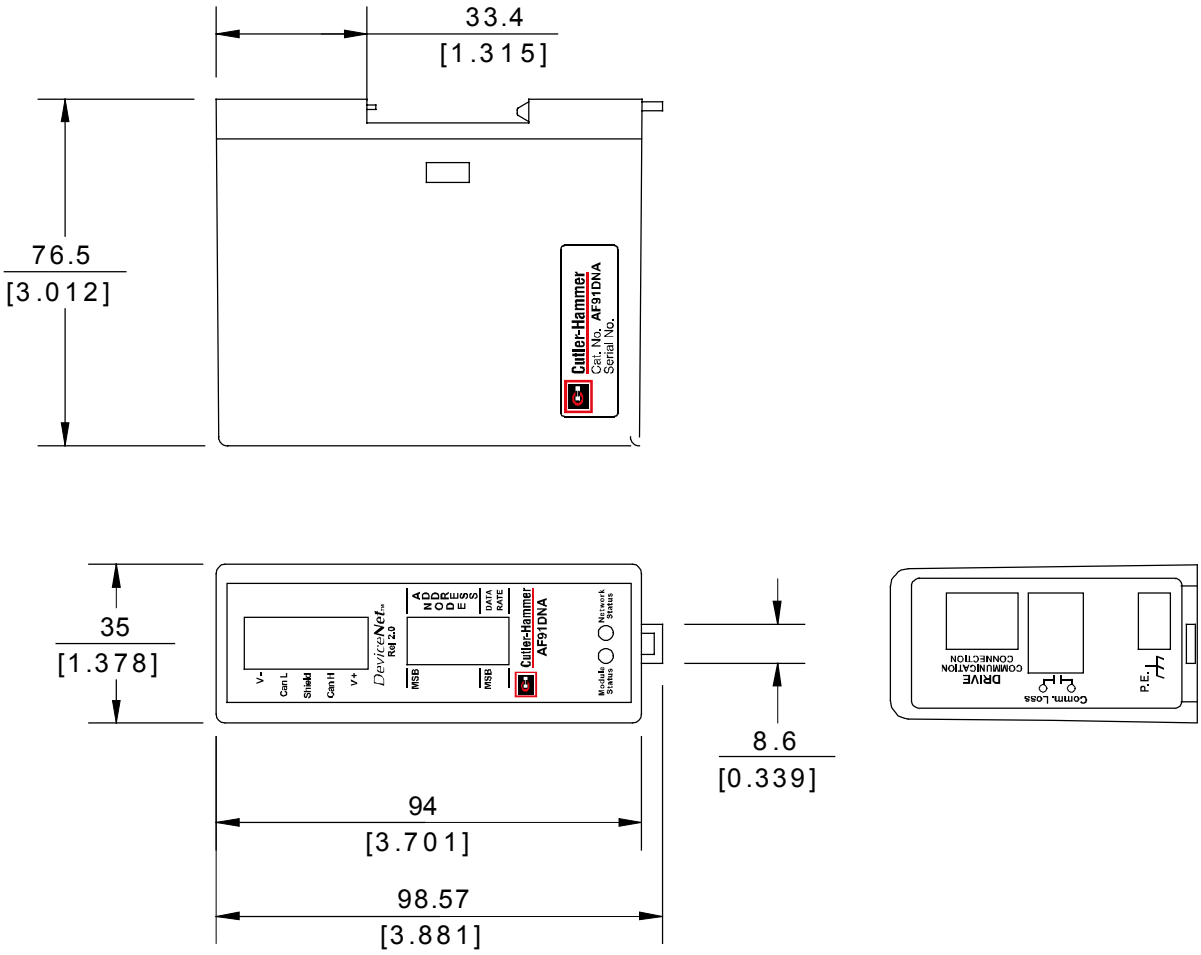
MECHANICAL

- Enclosure rating: NEMA 1
- Dimensions
  See drawings
- Screw torque:
  DeviceNet terminal block: 5 in-lb.
  Relay terminal block: 5 in-lb.

APPROVALS

- UL and cUL
- CE marked

AF91DNA Mounting Dimensions



AF91DNA Installation Instructions

⚠ Warning

DO NOT INSTALL OR PERFORM MAINTENANCE ON THIS DEVICE WHILE THE CONTROLLER IS ENERGIZED. DEATH OR SEVERE PERSONAL INJURY CAN RESULT FROM CONTACT WITH ENERGIZED EQUIPMENT. VERIFY THAT NO VOLTAGE IS PRESENT BEFORE PROCEEDING WITH INSTALLATION OR MAINTENANCE. Only qualified persons, as defined in the National Electric Code, should be permitted to install, maintain, or operate this device. These persons should be familiar with the installation, maintenance, and operation of this device and the equipment onto which it is to be installed. Additionally, these persons should be familiar with all applicable local, state, and national regulations, industry standards, and accepted practices regarding safety of personnel. These instructions are provided only as a general guide to such qualified persons and are not all-inclusive. They do not cover every application or circumstance which may arise in the installation, maintenance, or operation of this equipment. Users are advised to comply with all local, state, and national regulations, industry standards, and accepted practices regarding safety of personnel and equipment.

⚠ Warning

BEFORE INSTALLING AND OPERATING THE AF91DNA INTERFACE WITH AN AF91 INVERTER, YOU SHOULD BE FAMILIAR WITH THE WARNINGS AND CAUTIONS FOUND IN THE AF91 INSTRUCTION MANUAL. YOU SHOULD ALSO BE FAMILIAR WITH THE FEATURES AND FUNCTIONS OF THE INVERTER AS THEY APPLY TO YOUR APPLICATION.

NOTES:

1. Do not connect or disconnect the AF91DNA interface to a powered AF91 inverter. Doing so may result in an inverter trip.
2. The plastic package of the AF91DNA is not meant to be directly mounted to a panel. Damage may result if screws are used to attach the AF91DNA to a panel.

INSTALLATION:

1. Remove all power to the AF91 inverter and wait for all displays to extinguish.
2. Determine a mounting location for the AF91DNA interface that is within the reach of the cable to the AF91 inverter and away from heat generating components.
   DIN Rail Mounting:
   - Attach DIN rail, (Cutler-Hammer PN 383TS35) to panel.
   - Snap AF91DNA interface onto DIN rail.
   - For applications where vibration is a concern, secure the AF91DNA in place with DIN rail end stops (Cutler-Hammer PN C383ES35) on each side of the module.
3. Supplied with your AF91DNA is a length of wire which is terminated at one end by a quick disconnect type connector. Attach this connector to the PE tab at the bottom of the unit. Connect the other end to the grounded panel the AF91DNA is mounted on. To minimize the effects of electrical disturbances on the product, trim this wire to the shortest possible length before connecting to ground.

4. While in network control, if the Interface Cable between the AF91DNA DeviceNet Interface and the AF91 inverter is damaged or removed, drive communications will be lost. By default the AF91DNA configures the AF91 inverter to respond to this loss of communication by tripping and stopping the inverter. To enable this response the AF91DNA Comm Loss contact must be wired to the inverter as described below. Alternately the inverter can be configured so that a loss of communication automatically results in a Free Run Stop (FRS) or Hold Last Speed as described below.

   Fault/Stop – Loss of communications activates an External Trip (fault) which trips the inverter, removing power to the motor. The trip must be reset before the inverter can be restarted.

   Free Run Stop (FRS) – Loss of communications activates an AF91 Free Run Stop causing the inverter output to be inhibited, removing power to the motor. If communications is restored the inverter will again respond to messages from the AF91DNA.

   If the desired response is to Fault/Stop or Free Run Stop, wire the AF91DNA's Comm Loss contact, located at the bottom of the enclosure, to TB5 and P24 as shown below. For Free Run Stop you will also need to set the TB5-Comm Loss attribute in the AF91 Protections Object during configuration of the AF91DNA.

   ⚠ Warning

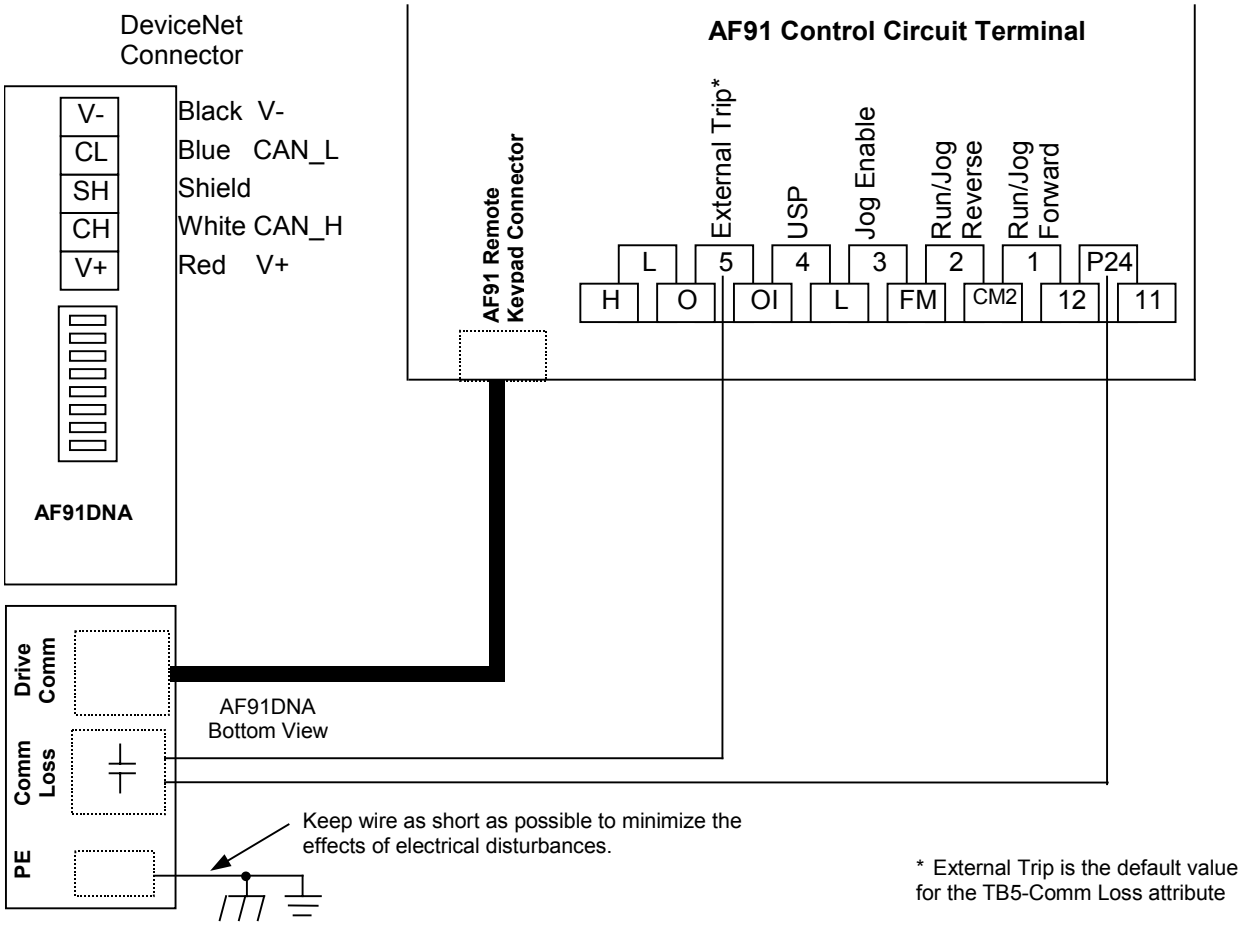
   If Run/Stop control was set to the AF91 terminal block (NetCtrl = 0 or Running Command Source Setting = 0) and communications is restored, the drive will automatically restart, applying power to the motor.

   ⚠ Warning

   With the Comm Loss contact connected to TB5 of the AF91 inverter, do not program the AF91's intelligent terminal 5 for any of the Run or Jog functions. Failure to observe this warning could result in the inadvertent starting of the drive upon loss of communications between the AF91DNA and the AF91 inverter.

Hold Last Speed – Loss of communications causes the drive to hold the last setting of Frequency Reference (Speed Reference) and continue to operate indefinitely until stopped by external means. If communications is restored the inverter will again respond to messages from the AF91DNA.

If the desired response is Hold Last Speed, the Comm Loss contact should not be wired to the inverter. However, an external Stop circuit must be provided to stop the drive when it is operating in the Hold Last Speed mode. Also, during configuration, the intelligent input terminal TB5 must be reprogrammed to enable normal operation.



5. If your application requires that the drive can be shut down by external means regardless of the mode of operation (Local, Remote, or Network), an external Inverter Emergency Stop must be installed. The E-Stop circuit should be connected to an AF91 input terminal that has been assigned the appropriate E-Stop function. Among your E-Stop choices are External Trip and Free Run Stop. Refer to the Section 7 of the AF91 Instruction Manual for a description of these functions and how to program them. Do not use terminal 5 if it is wired to the Comm Loss contact of the AF91DNA.
6. Wire DeviceNet to the terminals of the DeviceNet open connector at the top of the AF91DNA as shown above.

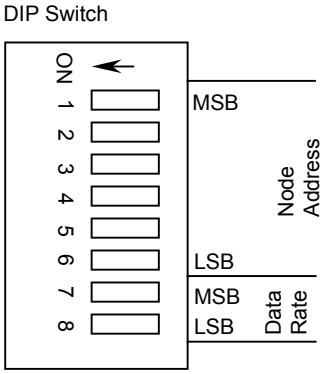
NOTE: Do not connect or disconnect the AF91DNA interface to the AF91 inverter while power is supplied to the inverter.

7. With the AF91 inverter unpowered, attach the interface cable to the RJ 45 connectors at the bottom of the interface and the bottom of the AF91 inverter.

⚠ Warning

The interface cable provided with the AF91DNA has an insulation rating of 300 volts. When routing this cable on your panel, it must not come in contact with the wiring of other circuits operating at voltages greater than 300 volts.

8. Determine the baud rate of your DeviceNet network and the desired Node Address of the AF91DNA. Set the DIP switch on the front of the interface accordingly. Note: The Node Address is the number set by switches 1 thru 6 where switch 1 is the most significant bit in this number. The baud rate is set using switches 7 and 8 as shown in the table below.



In the tables below, a "1" corresponds to the switch in the "ON" position.

Node Address	DIP Switch Position					
	1	2	3	4	5	6
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
...						
62	1	1	1	1	1	0
63	1	1	1	1	1	1

Baud rate bit/sec	DIP switch position	
	7	8
125K	0	0
250K	0	1
500K	1	0
N/A	1	1

Note

SINCE MANY NEW DEVICENET DEVICES ARE FACTORY SET TO NODE ADDRESS 63, IT IS USUALLY NOT A GOOD IDEA TO LEAVE A NODE ADDRESS SET AT 63. TWO NODES AT THE SAME NODE ADDRESS WILL CAUSE ONE OR BOTH NODES TO FAULT!

AF91DNA Configuration

⚠ Warning

Upon power-up, the AF91DNA initialization routine programs the function and polarity of terminal 5 of the AF91 inverter to the function specified by the TB5-Comm Loss attribute in the AF91 Protections Object (Class = 166 Instance = 1 Attribute ID = 14). This new function is not necessarily the same as the AF91's Cutler-Hammer out-of-box factory default for terminal 5. If the AF91 is to be returned to a stand alone, non-DeviceNet application, the function of terminal 5 must be reprogrammed to match the requirements of the new application. FAILURE TO DO SO COULD RESULT IN EQUIPMENT DAMAGE, PERSONAL INJURY, OR DEATH.

1. Apply power to the AF91 inverter. This will power up the interface and initiate the interface's power up sequence, as evidenced by the LED pattern displayed by the Module Status and Network Status LEDs.

   There must be a minimum of two powered nodes on the DeviceNet network for the Duplicate MAC ID check to complete successfully. The status of the check will be indicated by the MS and NS LEDs. See the LED Definition and Diagnostics section for the meaning of the LEDs. If the check fails, you will have to change the Node Address of one or more of your nodes.

2. If a drive fault is indicated on the AF91 Display (E12), reset it by pressing the STOP button on the keypad of the drive.

   If the Comm Loss contact is not wired to the AF91 inverter, temporarily jumper TB-5 to TB-P24 on the AF91's terminal block. (This jumper will be removed at the last step of this configuration procedure.) Now reset the fault by pressing the STOP button on the keypad of the AF91 inverter.

3. Using an appropriate DeviceNet configuration tool, set the attributes of the AF91DNA to implement your application. This may be accomplished by using the Cutler-Hammer NetView configuration tool. For more information on configuring nodes refer to the NetView User's Manual.

   The Table below lists the minimum set of attributes that MUST be set up during configuration of the AF91DNA.

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
AF91 Basic Parameters Class (160) - Instance (1)					
3 03_{hex}	**Base Frequency**	Get_Attribute_Single, Set_Attribute_Single	C7	60, 50, 360	AF91 Function A03 Sets the frequency at which inverter outputs rated voltage Actual value (Hz) = Base Frequency CANNOT be set during run state. Does not require EE Store Operation to save attribute in nonvolatile drive memory.
4 04_{hex}	**Maximum Frequency**	Get_Attribute_Single, Set_Attribute_Single	C7	60, 50, 360	AF91 Function A04 Sets the maximum inverter output frequency. During transient operation the output frequency may exceed the maximum by up to 1.5 Hz. Actual value (Hz) = Maximum Frequency CANNOT be set during run state. Does not require EE Store Operation to save attribute in nonvolatile drive memory.
12 $0C_{hex}$	**Local Reference Source**	Get_Attribute_Single, Set_Attribute_Single EE	C1	0, 0, 1	Specifies the source of the reference when NetRef transitions to zero. 0=AF91 Keypad (or POT) 1=Terminal Block

13 $0D_{hex}$	**Number of Poles**	Get_Attribute_Single, Set_Attribute_Single EE	C6	4, 2, 255	Specifies the number of poles in the connected motor. Used in speed calculations. Using data from an AC motor's nameplate: Motor Poles = (120*Hz)/RPM (rounded up to the nearest whole number) CANNOT be set during run state
14 $0E_{hex}$	**Motor Voltage**	Get_Attribute_Single, Set_Attribute_Single	C7	*, 200, 460	AF91 Function A82 Enter the nameplate voltage of the connected motor. Acceptable ranges: * 200 VAC Class: 200-240 (230 default) 400 VAC Class: 380-460 (460 default) Actual value (volts) = Motor Voltage CANNOT be set during Run state. Does not require EE Store Operation to save attribute in nonvolatile drive memory.
16 10_{hex}	**Polled Input Assembly Type**	Get_Attribute_Single, Set_Attribute_Single EE	C7	105, 105, 108	The input assembly used by the polled connection
17 11_{hex}	**Polled Output Assembly Type**	Get_Attribute_Single, Set_Attribute_Single EE	C7	100, 100, 101	The output assembly used by the polled connection
AF91 Protections Class (166) - Instance (1)					
11 $0B_{hex}$	**Safe State Behavior**	Get_Attribute_Single, Set_Attribute_Single EE	C6	0, 0, 2	Selects Safe State response to errors which specify safe state operation. Currently only a loss of connection other than by de-allocation is a safe state error. **Warning**: Review the application for safe operation before specifying a value for this attribute. 0 = Stop 1 = Hold Last Speed 2 = Preset Speed/Direction
12 $0C_{hex}$	**Safe State Preset Frequency**	Get_Attribute_Single, Set_Attribute_Single EE	C7	0, 0, 36000	Sets safe state frequency reference if the Safe State Behavior attribute specifies "Preset Speed/Direction". **Warning**: Review the application for safe operation before specifying a value for this attribute. Inverter will require external stop. Actual value (Hz) = (Safe State Preset Frequency)/100
13 $0D_{hex}$	**Safe State Preset Direction**	Get_Attribute_Single, Set_Attribute_Single EE	C6	2, 0, 3	Sets safe state direction of rotation if the Safe State Behavior attribute specifies "Preset Speed/Direction" . **Warning**: Review the application for safe operation before specifying a value for this attribute. Inverter will require external stop. 0 = Forward 1 = Reverse 2 = Hold Last Direction 3 = Go To Opposite Direction
14 $0E_{hex}$	**TB5-Comm Loss**	Get_Attribute_Single, Set_Attribute_Single EE	C6	2, 0, 2	Assigns FRS (Free Run Stop) or Ext Trip to AF91 input terminal 5 and sets polarity of terminal 5 to normally closed. 1=FRS assigned 2=Ext Trip assigned 0=Hold Last Speed. Uses customer assignment of input 5 or CH factory default if unassigned. Values 1 & 2 override any other assignment for input terminal 5.

In addition to the attributes listed above, Appendix A contains a complete list of the attributes supported by the AF91DNA.

- Attributes shown in **bold** type (also designated “EE” in the Access Rule column) are stored in the non-volatile memory of the AF91DNA Communication Interface and maintain their value after a power loss.

- Attributes that are *italicized* are stored in the non-volatile memory of the AF91 Drive and maintain their value after a power loss, but only if an EE Store Operation is performed prior to power down to save any attribute changes made since the last EE store.
- All other settable attributes will power up at their default values.
- Attributes listed in the shaded area must be set during the configuration of the AF91DNA while it is connected to an AF91 inverter. These attributes are stored in non-volatile memory and maintain their values after power loss.

Review the attribute list and configure the shaded area attributes as well as those that are required for your application.

Note: In order to start and stop the drive over DeviceNet, NetCtrl (Class = 168 Instance = 1 Attribute ID = 5) must be set to 1, "Network Control". Setting the speed of the drive over DeviceNet requires that NetRef (Class = 169 Instance = 1 Attribute ID = 4) be set to 1, "Network Reference".

4. There are several AF91 Protections Object attributes that you must pay special attention to. Set the following attributes for safe operation of the drive in your application.

⚠ Warning

BEFORE YOU CONFIGURE THE FOLLOWING ATTRIBUTES, REVIEW YOUR APPLICATION'S SAFETY REQUIREMENTS. FAILURE TO PROPERLY CONFIGURE THESE ATTRIBUTES COULD RESULT IN EQUIPMENT DAMAGE, PERSONAL INJURY, OR DEATH.

AF91 Protections Object Attributes (Class = 166 Instance = 1 Attribute = ID in table below)

ID	Name	
11	Safe State Behavior	Selects Safe State response to events that specify safe state operation.
12	Safe State Preset Frequency	Sets the Safe State speed if the Safe State Behavior attribute specifies "Preset Speed/Direction".
13	Safe State Preset Direction	Sets the Safe State direction of rotation if the Safe State Behavior attribute specifies "Preset Speed Direction".
14	TB5-Comm Loss	Determines response of drive to a loss of communications between drive and interface when the Comm Loss contact opens.

Safe State

AF91 Protections Object Attributes 11,12, & 13 specify the response of the drive to Safe State events. Currently only a Loss of DeviceNet Connection Other Than by Deallocation event is specified to have a Safe State response. Refer to this document's AF91DNA Control Supervisor section and Appendix A for more details.

Loss of Communications

As part of the configuration process, you must determine the desired response of the drive in the event communication between the AF91DNA interface and the AF91 inverter fails. The AF91 Protections Object Attribute TB5-Comm Loss allows you to select the response.

Your choices of values for this attribute are:

1 = **Free Run Stop** assigned to AF91 terminal 5. Overrides any other assignment for this input terminal.
2 = **External Trip** assigned to AF91 terminal 5. Overrides any other assignment for this input terminal.
0 = **Hold Last Speed**. Uses customer assignment of input 5 or CH factory default if unassigned.

Hold Last Speed requires that terminal 5 of the AF91 inverter be assigned a function other than FW (Forward Run/Stop), RV (Reverse Run/Stop), or JG (Jog). This is accomplished by setting AF91 parameters C05 and C15 using the procedure described in the AF91 Inverter Configuration section of this document and section 7, "Function of Control Circuit Terminals", of the AF91 Instruction Manual for more details.

⚠ Warning

If you chose the **Free Run Stop** (FRS) option for the Communications Loss shut down mode and Run/Stop control is set to the AF91 terminal block (NetCtrl = 0 or Running Command Source Setting = 0) when communications is restored, the drive will automatically restart, applying power to the motor.

5. When all attributes have been configured, perform an EEPROM Store by setting the EE Store Operation attribute (Class = 160 Instance = 1 Attribute ID = 15) to a value of 1.
6. When the store operation is complete, remove the jumper from step 2a (if necessary).

AF91 Inverter Configuration

Your application may require setting some of the AF91's parameters that not accessible over the DeviceNet network. Changes to these parameters can be made using the AF91's Digital Operator Keypad, but the AF91DNA must first be disconnected as noted below. Consult the AF91 Instruction Manual for additional details.

Notes:

1. **AF91DNA must be removed to program AF91 parameters from keypad**: Except for the STOP key, all keys of the Digital Operator Keypad are disabled while the AF91DNA is connected to the AF91. Access to the AF91's parameters requires that the (RJ45) Interface Cable be removed. Always remove power to the AF91 and wait for all displays to extinguish before removing or attaching this cable.
2. **Terminals 5 and 24 may need to be temporarily jumpered to clear E12 trips**: In most applications Terminal 5 of the AF91 will be configured by the AF91DNA to fault the drive (E12, External Trip) if this terminal is not tied to +24 VDC by the Comm Loss contact of the AF91DNA. If you are experiencing E12 trips while you are attempting to set AF91 parameters, temporarily jumper the AF91 terminals 5 and P24 together.
3. **AF91 Keypad Operation**: For a description of the inverter's keypad operation refer to section 8, "Operation of the Digital Operator Keypad", of the AF91 Instruction Manual.

LED Definitions and Diagnostics

The AF91DNA includes two LED DeviceNet status indicators: Module status and Network status.

When power is applied an LED test is performed, allowing verification of LED operation.

After the LED test and product initialization has been completed, the LEDs indicate the status shown in the tables below.

MODULE STATUS LED

LED	Meaning
Off	There is no power applied to the AF91DNA.
Green	The AF91DNA is operating normally.
Flashing Green	The AF91DNA is in the Standby state, or the device needs commissioning due to configuration missing, incomplete or incorrect
Flashing Red	The AF91DNA has detected a Recoverable Fault
Red	- The AF91DNA has detected an Unrecoverable Fault and may need replacing. - The Data Rate DIP Switches were both in the ON position when power was applied.

NETWORK STATUS LED

LED	Meaning
Off	AF91DNA is not on–line. - The device has not completed the Dup_MAC_ID test yet. - If the Module Status LED is off, the device is not powered.
Flashing Green	The AF91DNA has passed the Dup_MAC_ID test, is on–line, but is not allocated to a master.
Green	The AF91DNA is on–line and allocated to a Master.
Flashing Red	I/O Connection is in the Timed–Out state.
Red	The AF91DNA cannot communicate on the network (Duplicate MAC ID Check failure, or Bus–off). There may be another DeviceNet node with the same MAC ID.

AF91DNA Operation Notes

1. Restricted Attributes:
   The table below lists "restricted" attributes, attributes that are not allowed to be changed while the drive is running. These attributes should only be set when the drive is stopped.

Object Class (Class Code)	Instance 1 Attribute ID	Attribute Name
AF91 Basic Parameters (160)	1	Frequency Source Setting *
AF91 Basic Parameters (160)	2	Running Command Source Setting *
AF91 Basic Parameters (160)	3	Base Frequency
AF91 Basic Parameters (160)	4	Maximum Frequency
AF91 Basic Parameters (160)	9	Carrier Frequency Setting
AF91 Basic Parameters (160)	13	Number of Poles
AF91 Basic Parameters (160)	14	Motor Voltage *
AF91 I/O Signals (161)	1	Selection of PID Control
AF91 I/O Signals (161)	5	Scale Conversion of PID Control Setting
AF91 I/O Signals (161)	6	Feedback Source Setting
AF91 I/O Signals (161)	8	Level of Overload Signal Setting
AF91 I/O Signals (161)	9	Level of Deviation Signal Setting
AF91 Protections (166)	1	Selection of DC Braking
AF91 Protections (166)	2	DC Braking Frequency Setting
AF91 Protections (166)	3	DC Braking Waiting Time Setting
AF91 Protections (166)	4	DC Braking Force Setting
AF91 Protections (166)	5	DC Braking Time Setting
AF91 Protections (166)	6	Jump Frequency 1 Setting
AF91 Protections (166)	7	Jump Frequency Width 1 Setting
AF91 Protections (166)	8	Level of Electronic Thermal Setting
AF91 Protections (166)	9	Overload Restriction Level Setting
AF91 Protections (166)	10	Selection of Operation at FRS Signal Canceled

* Changes to these attributes are not "restricted" by the AF91DNA. Although it is possible, these attributes should not be changed while the drive is running.

2. Reset Service:
   The following table lists the different types of resets supported by the Identity Object. Sending a reset of any type is not permitted while the AF91 drive is running. Reset types 1, 100, and 101 should be a "last resort" to recover from an unknown configuration. Resetting the drive will change any reassignment of the intelligent terminal block functions. Make sure you verify, and reconfigure if necessary, the terminal block assignments before starting the drive.

⚠ Warning

Resetting the AF91DNA interface to its out-of-box configuration will set ALL attributes to their Cutler-Hammer default values and change the response of the drive to a loss of communications with the AF91DNA. The device will have to be re-configured for your application before resuming normal operation. FAILURE TO DO SO COULD RESULT IN EQUIPMENT DAMAGE, PERSONAL INJURY OR DEATH

Resetting the AF91 inverter to its out-of-box configuration will set ALL inverter parameters to their Cutler-Hammer default values. The assignment of the AF91's intelligent terminal block functions will be changed to their default functions which could result in abnormal operation and/or loss of protection functions. Before restarting the inverter, you must verify that it is properly configured for your application. FAILURE TO DO SO COULD RESULT IN EQUIPMENT DAMAGE, PERSONAL INJURY OR DEATH.

Value:	Type of Reset:
0	Emulate as closely as possible the cycling of power to the AF91DNA DeviceNet Interface. This value is the default if this parameter is omitted.
1	Return the AF91DNA DeviceNet Interface AND the AF91 Drive as closely as possible to the out–of–box (C-H factory default) configuration, then emulate cycling of power as closely as possible.
100	Return the AF91DNA DeviceNet Interface as closely as possible to the out–of–box (C-H factory default) configuration, then emulate cycling of power as closely as possible.
101	Return the AF91 Drive as closely as possible to the out–of–box (C-H factory default) configuration, then emulate cycling of power as closely as possible.

3. The AF91 inverter has a timer that accumulates the time of operation of the inverter. This time of operation is used by the AF91DNA to time stamp inverter trips in the AF91 Trip History Object (Class = 167 Instance = 1 Attribute IDs = 3, 7, & 11). Since the timer is not recording continuous time, any time stamp value is relative to the actual inverter's operating time since it was initialized at the factory.
4. An attempt to set attributes beyond their min/max limits will not be executed. Explicit messaging will respond with an Invalid Attribute Value error (09). Out of range Assembly Object attributes will not generate an error.
5. Output Assembly attribute values of the active output assembly which are set by explicit messaging will be overwritten the next time a poll command is received with new data for the respective attribute value.
6. Drive Faults: If a drive fault occurs, it should be reset over the DeviceNet network. (Pressing the Stop button on the AF91 inverter clears the fault in the AF91 only. It does not reset the AF91DNA's fault logic.) Resetting the drive fault over the network requires a 0 to 1 Transition of FaultRst (Class = 168 Instance = 1 Attribute ID = 12).

   Note: In network control, the drive will restart after a drive fault when the fault has been reset AND Run1 (or Run2) has been set to 1 again, either by an explicit message or by the next polled I/O message.
7. The AF91DNA is designed to RUN whenever Run1 or Run2 (Class = 168 Instance = 1 Attribute ID = 3 or 4) is set while CtrlFromNet (Class = 168 Instance = 1 Attribute ID = 15) is set and Faulted (Class = 168 Instance = 1 Attribute ID = 10) is not set. This is different from some other DeviceNet motor control products that require a 0 to 1 transition on Run1 or Run2. A faulted AF91DNA will start running upon a 0 to 1 transition of FaultRst (Class = 168 Instance = 1 Attribute ID = 12) if Run1 and CtrlFromNet are set. An AF91DNA will start running upon a 0 to 1 transition of NetCtrl if Run1 is set and Faulted is not set.
8. It is strongly recommended that NetCtrl (Class = 168 Instance = 1 Attribute ID = 5) always be used instead of Running Command Source Setting (Class = 160 Instance = 1 Attribute ID = 2) to avoid any confusion which may be caused by the interdependence between the two attributes.
9. It is also strongly recommended that NetRef (Class = 169 Instance = 1 Attribute ID = 4) and Local Reference Source (Class = 160 Instance = 1 Attribute ID = 12) always be used instead of Frequency Source Setting (Class = 160 Instance = 1 Attribute ID = 1) to avoid any confusion which may be caused by the interdependence between the attributes.
10. SpeedRef (RPM) (Class = 169 Instance = 1 Attribute ID = 8) is the preferred attribute to use for setting the speed of the drive. Speed can also be set using Output Frequency Reference (Hz) (Class = 160 Instance = 1 Attribute ID = 5) but this value will be overwritten by the next I/O or explicit message which sets SpeedRef.
11. Many AF91 parameters are mapped as attributes in the AF91DNA application objects, Classes 160 thru 170. These parameters may be accessed via explicit messaging. An explicit response may occur 800 milliseconds or later after its request.

12. At lease 30 seconds should be allowed for an Identity Reset, Type 1.
13. The default values for Frequency Source Setting (Class = 160 Instance = 1 Attribute ID = 1), Running Command Source Setting (Class = 160 Instance = 1 Attribute ID = 2), and Overload Restriction Level Setting (Class = 166 Instance = 1 Attribute ID = 9) are dependent on the model and date code of the AF91 inverter. The user should ensure the desired value rather than relying on published default values.
14. During the EE Store Operation the Running Command Source Setting attribute (Class = 160 Instance = 1 Attribute ID = 2) is automatically set = 2 (network). This is done to guarantee that the drive cannot be started from the AF91 input terminals while the store is occurring.
15. The AF91DNA relies on the AF91 inverter to write default values to its parameters during “Data initialization” (b84 = 1) and Identity Resets, Type 1 and 101.

AF91DNA Troubleshooting

1) No LED's on the AF91DNA interface module.
   a) AF91 may not be powered
   b) Communication cable may be damaged or not connected properly
   c) Drive may not be configured as a Cutler-Hammer AF91. See the AF91 Instruction Manual – functions b84 & b85.
2) NS LED solid red
   a) Incorrect baud rate for network
   b) Check for another device with same MAC ID
3) NS LED off
   a) No devices are communicating on the network
4) The drive will not start
   a) Verify that the NetCtrl bit is set high in the output assembly and the CtrlFromNet bit is high in the input assembly
   b) Verify that the NetRef bit is set high in the output assembly and that a speed reference is given
   c) Verify that the TB5 connection and configuration is correct for your application
5) The motor is at a speed other then is being sent to it from the output assembly
   a) Verify that you have set the number of motor poles correctly
   b) Verify that the NetRef bit is set high
6) Drive Replacement: In the event that the AF91 drive is replaced, the customer will must verify that the intelligent I/O terminals are correctly programmed for the intended application. The procedure to restore the terminal block functionality is:
   a) Verify the TB5-Comm Loss attribute is correct.
   b) Reset the drive over the network to properly program TB-5.
   c) Manually program the other intelligent inputs (1-4) and outputs (11-12) to the required functions using the procedure outlined in the AF91 Instruction Manual.
7) Drive does not respond to new attribute value setting;
   a) Attribute value exceeds min/max limit.

AF91DNA DeviceNet Interface Errors

The following errors may be sent as error responses over DeviceNet. They are not logged in the Event List attributes of the AF91 Trip History Object.

AF91DNA DeviceNet Errors	dec	hex	dec	hex		
Error Name	General Error Code		Add'l Error Code		DeviceNet Description	Additional Description
Resource unavailable	2	02_{hex}			Resources needed for the object to perform the requested service were unavailable	
Service not supported	8	08_{hex}			The requested service was not implemented or was not defined for this Object Class/Instance.	
Invalid attribute value	9	09_{hex}			Invalid attribute data detected	Data not within the specified valid data range
Already in requested mode/state	11	$0B_{hex}$			The object is already in the mode/state being requested by the service	
Object state conflict	12	$0C_{hex}$			The object cannot perform the requested service in its current mode/state	Includes: Attribute not settable while drive is in Running state
Attribute not settable	14	$0E_{hex}$			A request to modify a non-modifiable attribute was received.	
Device state conflict	16	10_{hex}			The device's current mode/state prohibits the execution of the requested service.	
Reply data too large	17	11_{hex}			The data to be transmitted in the response buffer is larger than the allocated response buffer	
Not enough data	19	13_{hex}			The service did not supply enough data to perform the specified operation.	
Attribute not supported	20	14_{hex}			The attribute specified in the request is not supported	
Too much data	21	15_{hex}			The service supplied more data than was expected	
Object does not exist	22	16_{hex}			The object specified does not exist in the device.	
No stored attribute data	24	18_{hex}			The attribute data of this object was not saved prior to the requested service.	
Store operation failure	25	19_{hex}			The attribute data of this object was not saved due to a failure during the attempt.	
Drive Communication Error	31	$1F_{hex}$	1	01_{hex}	Inverter interface communication error	Indicates communication error with the drive.
Connection object state conflict	31	$1F_{hex}$	19	13_{hex}	Polled I/O connection is in the Established state.	Assembly type cannot be set.
Invalid parameter	32	20_{hex}			A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.	
Invalid Member ID	40	28_{hex}			The Member ID specified in the request does not exist in the specified Class/Instance/Attribute	

The following is a description of the codes that may be logged in the Event List attributes of the AF91 Trip History Object.

AF91DNA Events	dec	hex		
Event Name	Event Code		Event Description	Additional Description
No event	0	00_{hex}		Default value in Event List entries
Drive Communication Error	1	01_{hex}	Inverter interface communication error	Indicates communication error with the drive.
I/O Connection Timeout – Fault Stop	2	02_{hex}	Uncontrolled loss of DeviceNet communication.	Transition to Fault_Stop.
I/O Connection Timeout – Hold Last Speed	3	03_{hex}	Uncontrolled loss of DeviceNet communication.	Remain in Enabled State. Hold Last Speed
I/O Connection Timeout – Run at Preset Speed and Direction	4	04_{hex}	Uncontrolled loss of DeviceNet communication.	Remain in Enabled State. Run at Preset Speed & Direction
Low DeviceNet voltage – Fault Stop	5	05_{hex}	Uncontrolled loss of DeviceNet communication.	Transition to Fault_Stop.
Low DeviceNet voltage – Hold Last Speed	6	06_{hex}	Uncontrolled loss of DeviceNet communication.	Remain in Enabled State. Hold Last Speed
Low DeviceNet voltage – Run at Preset Speed and Direction	7	07_{hex}	Uncontrolled loss of DeviceNet communication.	Remain in Enabled State. Run at Preset Speed & Direction
Bus-off – Fault Stop	8	08_{hex}	Uncontrolled loss of DeviceNet communication.	Transition to Fault_Stop.
Bus-off – Hold Last Speed	9	09_{hex}	Uncontrolled loss of DeviceNet communication.	Remain in Enabled State. Hold Last Speed
Bus-off – Run at Preset Speed and Direction	10	$0A_{hex}$	Uncontrolled loss of DeviceNet communication.	Remain in Enabled State. Run at Preset Speed & Direction
CAN overrun – Hold Last Speed	12	$0C_{hex}$	Loss of DeviceNet message.	Remain in Current State. Hold Last Speed
Configuration Consistency Value (CRC) mismatch	14	$0E_{hex}$	Device's configuration is incorrect or incomplete	Major Recoverable Fault. An Identity Reset type 1 or 100 is needed for recovery
Microprocessor watchdog timeout	15	$0F_{hex}$	The device detected a serious problem with itself	Major Unrecoverable Fault
Received explicit message is too big	16	10_{hex}		
Received IO message is too big	17	11_{hex}		
Inverter voltage class out of range	18	12_{hex}	Voltage class of AF91 inverter is not 200V or 400V.	
Connection object state conflict	19	13_{hex}	Polled I/O connection is in the Established state.	Assembly type cannot be set.
I/O connection released – Stop	20	14_{hex}	Loss of DeviceNet I/O connection by de-allocation.	Transition to Stopping State.
Receive_Idle – Stop	23	17_{hex}	Loss of I/O data by Receive_Idle event.	Transition to Stopping State.
Receive_Idle – Hold Last Speed	24	18_{hex}	Loss of I/O data by Receive_Idle event.	Remain in Enabled State. Hold Last Speed
Explicit Connection Released – Stop	26	$1A_{hex}$	Loss of DeviceNet I/O connection by de-allocation.	Transition to Stopping State.
Receive Queue Overrun	32	20_{hex}	Loss of DeviceNet message.	Remain in Current State. Hold Last Speed
Transmit Queue Overrun	33	21_{hex}	Loss of DeviceNet message.	Remain in Current State. Hold Last Speed

Note: AF91 Trips are reset by pressing the STOP key on the AF91 Keypad or by a 0→1 transition of the FaultRst attribute (Class = 168 Instance = 1 Attribute ID = 12)

AF91 AFD Trips (Refer to Section 9 of the AF91 Instruction Manual)	AF91 Display Trip Code	**AF91DNA Trip Codes** AF91 Trip History Object Attribute: Cause of Trip n Class=167 Instance=1 Attribute=4, 8, or 12				
		Operating mode (All values are in hex)				
		Constant Speed	Decel	Accel	Other	
Overcurrent during constant speed	E01	C0-C2, FA-FD				Inverter output current has exceeded a predetermined level
Overcurrent during deceleration	E02		80-82, BA-BD			Inverter output current has exceeded a predetermined level
Overcurrent during acceleration	E03			40-42, 7A-7D		Inverter output current has exceeded a predetermined level
Overcurrent during other conditions	E04				00-02, 3A-3D	Inverter output current has exceeded a predetermined level
Overload	E05	C5, E0	85, A0	45, 60	05, 20	Motor overload has been detected by electronic thermal function
Overvoltage	E07	C3, FE	83, BE	43, 7E	03, 3E	DC bus voltage has exceeded predetermined level
EEPROM error	E08	C6, F9	86, B9	46, 79	06, 39	- Memory error due to noise or excessive temperature - May occur on power-up of inverter if certain inverter parameters were previously changed but an EE Store Operation was not performed before inverter power was removed.
Undervoltage	E09	C4, DF, FF	84, 9F, BF	44, 5F, 7F	04, 1F, 3F	DC bus voltage has dropped below predetermined level
CPU error	E11 / E22	C8, CE, E8-EF, F0-F2, F4	88, 8E, A8-AF, B0-B2, B4	48, 4E, 68-6F, 70-72, 74	08, 0E, 28-2F, 30-32, 34	Malfunction of CPU
External trip	E12	E5	A5	65	25	The AF91 intelligent terminal assigned the External Trip has sensed an active input signal. See the TB5-Comm Loss attribute in the AF91 Protections Object. Possible causes: - Loss of communications with AF91DNA DeviceNet Interface (AF91DNA Comm Loss contact open) - AF91DNA Comm Loss contact not wired to AF91 terminal block.
USP error	E13	E1	A1	61	21	With the USP function enabled, power was applied with one of the Run inputs active
Ground fault	E14	CD	8D	4D	0D	Ground current between the inverter and the motor has exceeded a predetermined level
Input overvoltage	E15	CF	8F	4F	0F	Input voltage has exceeded a predetermined level
Thermal	E21	CA	8A	4A	0A	Inverter module temperature has exceeded a predetermined level
PTC error	E35	E2	A2	62	22	External thermistor resistance has exceeded a predetermined value

AF91DNA Control Supervisor

INTRODUCTION

The Control Supervisor provides Run/Stop control logic between a set of DeviceNet connections and a motor in a Motor Control Device. A number of attributes may be accessed via I/O assemblies or explicit messaging. This description is an extension to the ODVA DeviceNet specification. It defines behaviors that may be implied but are not specifically stated in the ODVA document.

DEFINITIONS

These definitions describe events and variables used later in this description.

IO_Timeout - an event which indicates that the Polled I/O connection has timed out and has transitioned from the Established state.

Exp_Timeout - an event which indicates that the Explicit connection has timed out and has transitioned from the Established state.

IO_Released - an event which indicates that the Polled I/O connection has been released by request of the network master.

Exp_Released - an event which indicates that the Explicit connection has been released by request of the network master.

Receive_Idle - an event which indicates that the Polled I/O connection received zero-length data.

Power_Loss - an event which indicates that DeviceNet power has dropped below a minimum voltage.

Bus_Off - an event which indicates that the DeviceNet CAN controller has entered the Bus Off state.

Fault_Reset - an event corresponding to the transition of the value of the FaultRst attribute from 0 to 1.

DriveFault - an event which indicates that the drive has detected a problem.

ExpCommTimeout - a complex event created from Exp_Timeout and I/O connection state.

CommReleased - a complex event created from IO_Released, Exp_Released, and I/O connection state.

ReceiveIdleStop - a complex event created from Receive_Idle and Idle_State_Behavior.

SafeFault - a complex event created from CtrlFromNet, a "communication loss" event, and Safe State Behavior.

SafeChange - a complex event created from CtrlFromNet, a "communication loss" event, and Safe State Behavior.

IdleMode - a Boolean variable that is True when the Polled I/O connection is established and is receiving zero-length data.

FwdMode - a Boolean variable that is True when the drive is commanded to forward and the drive State is Enabled.

RevMode - a Boolean variable that is True when the drive is commanded to reverse and the drive State is Enabled.

Run1Var - a Boolean variable that contains the last received value of the Run1 attribute (read by Explicit Get of Run1).

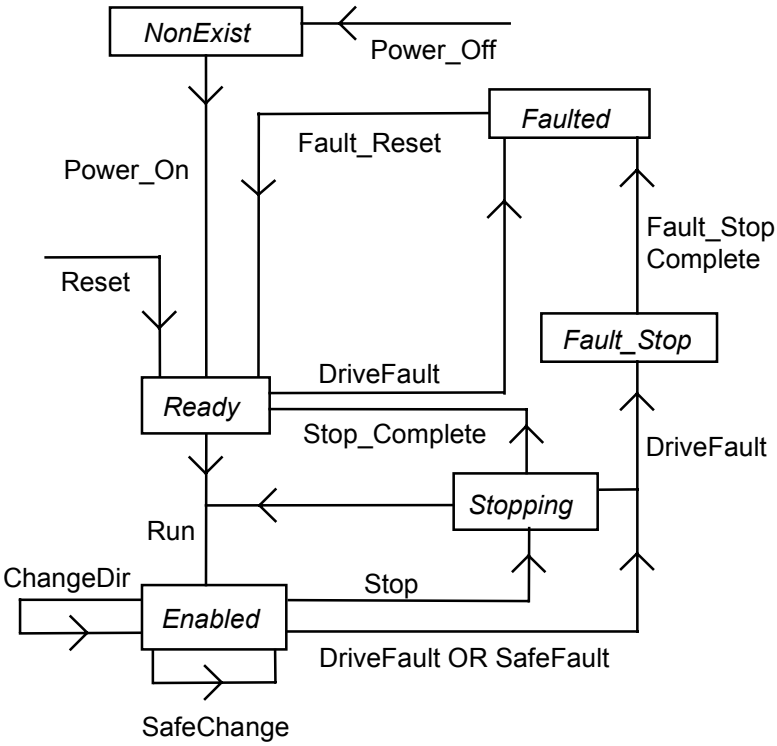
Run2Var - a Boolean variable that contains the last received value of the Run2 attribute (read by Explicit Get of Run2).

CONTROL SUPERVISOR BEHAVIOR

Control Supervisor State Transition Diagram

The State Transition Diagram provides a graphical description of the states and corresponding state transitions. This diagram is derived from a similar diagram in the ODVA DeviceNet specification. Names of state transitions are kept simple for clarity. Consult the Control Supervisor State Transition Table for detailed explanations of state transitions.

Control Supervisor State Transition Diagram



Control Supervisor State Transition Table

The State Transition Table describes the requirements and actions for each state transition.

0 = False, not active
1 = True, active
x = don't care

Important Facts:
(1) If the I/O connection is in the Established state while Control Supervisor is in the Enabled state, then the 5 request bits of the Output Assembly (Run1, Run2, FaultRst, NetCntrl, and NetRef) are being supplied by the I/O connection. Any writes to these bits by Explicit Requests would be overwritten by I/O Poll Commands. The I/O Poll Commands must contain data because Receive_Idle causes a transition to the Stopping state. Therefore, Receive_Idle or loss of the I/O connection is treated as a loss of the source of those request bits (even if the Explicit connection is still in the Established state). In other words, if an established I/O connection exists, that I/O connection is usually considered as the source of any control or reference information from DeviceNet.
(2) Running1 and Running2 are based on the running status and direction of the motor. Only one can be True at any time. They are not dependent on CtrlFromNet or the state of the Control Supervisor.

```
 ExpCommTimeout = (Exp_Timeout) AND (IO_Cnxn_State <> Established);
    CommReleased = (IO_Released) OR ( (Exp_Released) AND (IO_Cnxn_State <> Established) );
   ReceiveIdleStop = (Receive_Idle) AND (Idle_State_Behavior = 0);
        CtrlFromNet = (NetCntrl) AND (RunningCommandSourceSetting = Network);
          Run(Fwd) = (CtrlFromNet) AND (Run1Var) AND (NOT Run2Var) AND (NOT IdleMode);
          Run(Rev) = (CtrlFromNet) AND (NOT Run1Var) AND (Run2Var) AND (NOT IdleMode);
    ChangeDir(Fwd) = (CtrlFromNet) AND (Run1Var) AND (NOT Run2Var) AND (RevMode);
    ChangeDir(Rev) = (CtrlFromNet) AND (NOT Run1Var) AND (Run2Var) AND (FwdMode);
              Stop = (CtrlFromNet) AND
                     ( ( (NOT Run1Var) AND (NOT Run2Var) ) OR
                       (ExpCommTimeout) OR
                       (CommReleased) OR
                       (ReceiveIdleStop) );
         SafeFault = (CtrlFromNet) AND (DNFaultMode = 0) AND ( (IO_Timeout) OR (Power_Loss) OR (Bus_Off) );
       SafeChange = (CtrlFromNet) AND (DNFaultMode = 2) AND ( (IO_Timeout) OR (Power_Loss) OR (Bus_Off) );
```

Input Conditions								Results	
Old State	**Ctrl From Net**	**Run1 Var**	**Run2 Var**	**Idle Mode**	**Fwd Mode**	**Rev Mode**	**Event**	**New State**	**Action**
x	x	x	x	x	x	x	Power_Off	NonExist	
x (except NonExist)	x	x	x	x	x	x	Reset	Ready	Faulted := 0 FwdMode := 0; RevMode := 0 Run1Var := 0; Run2Var := 0
NonExist	x	x	x	x	x	x	Power_On	Ready	Faulted := 0 FwdMode := 0; RevMode := 0 Run1Var := 0; Run2Var := 0
Ready	x	x	x	x	x	x	DriveFault	Faulted	Faulted := 1
Ready	1	1	0	0	x	x	Run (Fwd)	Enabled	FwdMode := 1 (Start Forward)
Ready	1	0	1	0	x	x	Run (Rev)	Enabled	RevMode := 1 (Start Reverse)
Enabled	x	x	x	x	x	x	DriveFault	Fault_Stop	Faulted := 1 (Initiate Faulted Stop) FwdMode := 0; RevMode := 0 Run1Var := 0; Run2Var := 0
Enabled	1	x	x	x	x	x	Stop (CommRelease)	Stopping	Run1Var := 0; Run2Var := 0 (Initiate Stop)
Enabled	1	x	x	x	x	x	Stop (Receive_Idle)	Stopping	Run1Var := 0; Run2Var := 0 (Initiate Stop)
Enabled	1	x	x	x	x	x	Stop (PowerLoss)	Stopping	Run1Var := 0; Run2Var := 0 (Initiate Stop)
Enabled	1	0	0	x	x	x	Stop	Stopping	(Initiate Stop)
Enabled	1	1	0	0	0	1	ChangeDir (Fwd)	Enabled	FwdMode := 1; RevMode := 0 (Change to Forward)
Enabled	1	0	1	0	1	0	ChangeDir (Rev)	Enabled	FwdMode := 0; RevMode := 1 (Change to Reverse)
Enabled	1	x	x	x	x	x	SafeFault	Fault_Stop	Faulted := 1 (Initiate Faulted Stop) FwdMode := 0; RevMode := 0 Run1Var := 0; Run2Var := 0
Enabled	1	x	x	x	x	x	SafeChange	Enabled	FwdMode := Run1Var := (per Preset Direction) RevMode := Run2Var := (per Preset Direction) SpeedRef := (per Preset Frequency)
Stopping	x	x	x	x	x	x	DriveFault	Fault_Stop	Faulted := 1 (Initiate Faulted Stop)
Stopping	1	1	0	0	x	x	Run (Fwd)	Enabled	FwdMode := 1 (Start Forward)
Stopping	1	0	1	0	x	x	Run (Rev)	Enabled	RevMode := 1 (Start Reverse)
Stopping	x	0	0	x	x	x	Stop_Complete	Ready	
Fault_Stop	x	x	x	x	x	x	Fault_Stop Complete	Faulted	
Faulted	x	x	x	x	x	x	Fault_Reset	Ready	Faulted := 0

Other logic equations:

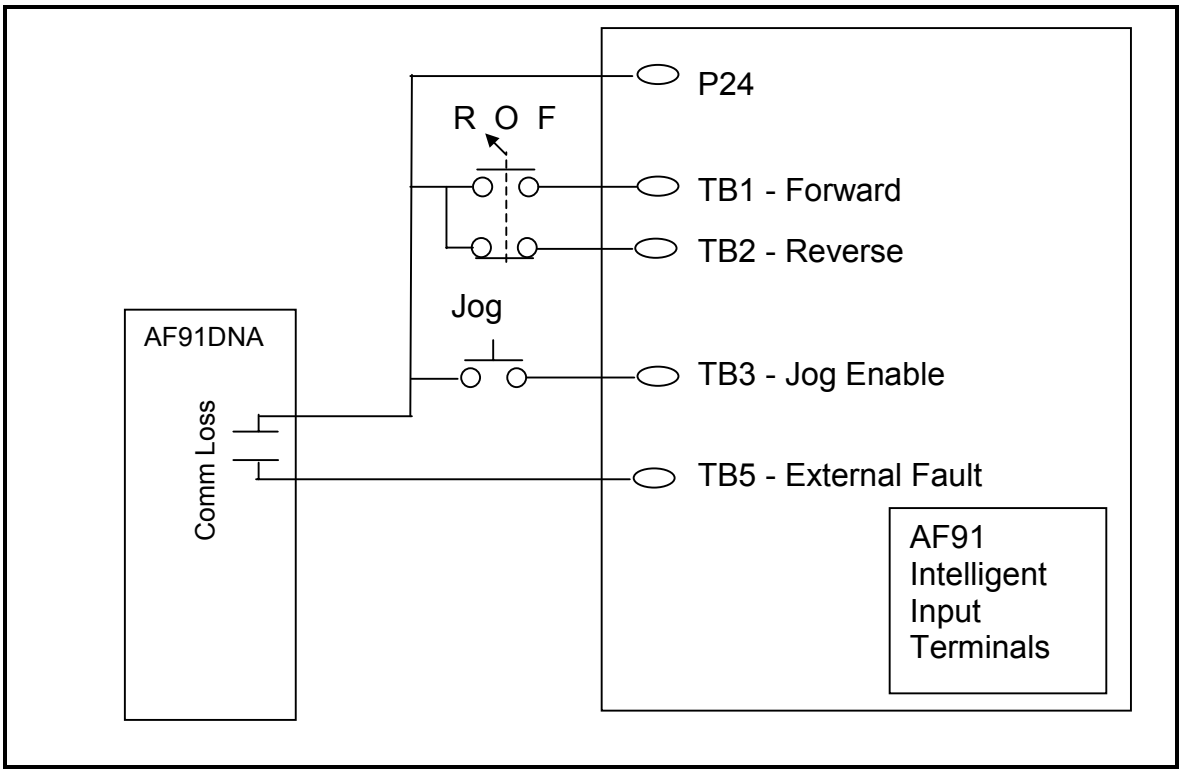
RefFromNet = (NetRef) AND (Frequency_Source_Setting = Network);

IF (RefFromNet)
{
(Write reference frequency to the drive whenever SpeedRef or Output_Frequency_Reference are written.)
}

When performing changes to achieve programmed Safe State:
(1) Run/Stop/Direction can be changed because CtrlFromNet must equal 1 when in the Enabled state.
(2) Reference can be changed to Preset Speed only if (RefFromNet = 1).

Hand - Auto (HA) Application Note

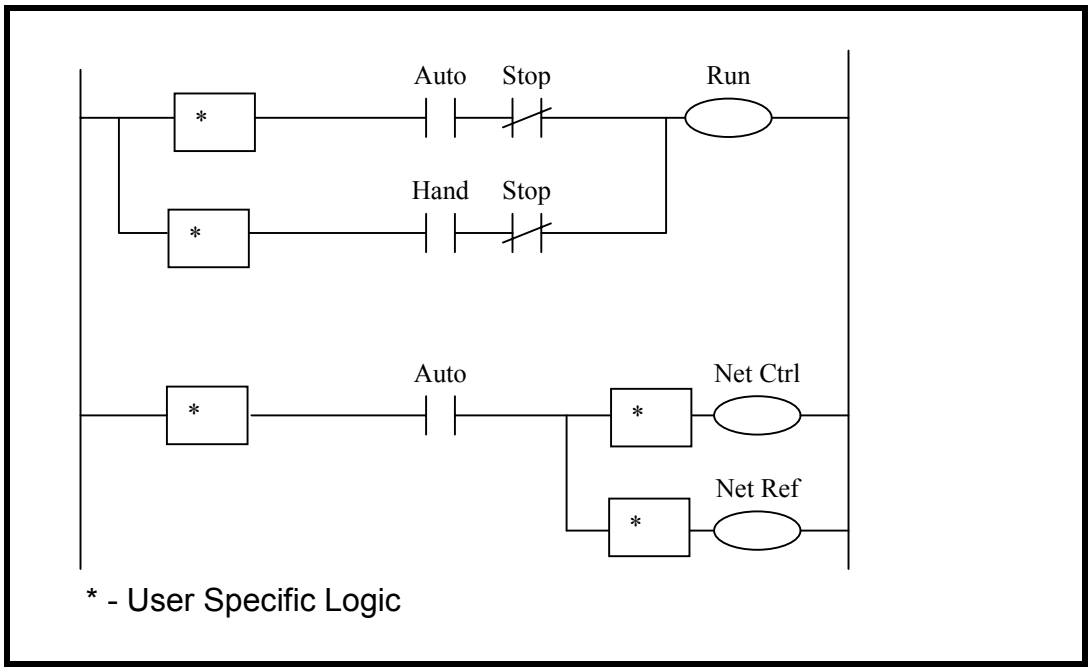
The wiring diagram below shows the AF91 configured for Hand-Auto operation.

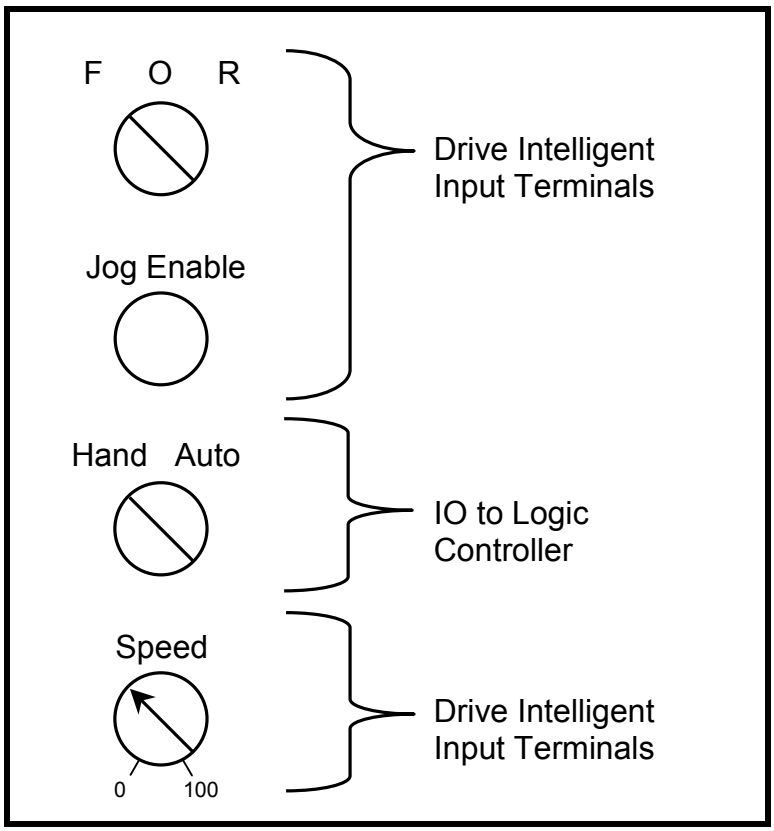


- Jog requires Jog Enable to be high prior to selection of direction.
- To control the state of the drive in Hand, stop the drive and change the output assembly NetCtrl bit from 1 to 0. At this point the AF91's intelligent input terminals will become active. Setting the NetCtrl bit back to 1 (Auto), deactivates the terminals and returns control to the network.
- To manually control the speed of the drive in Hand or Automatic, change the NetRef bit from 1 to 0. The Local speed pot will become active. Changing NetRef back to 1 will restore control to the output assembly's Speed Reference.

Note: Whenever changing the state of the NetCtrl or the NetRef bits within the output assembly, the drive should be in a Stopped state.

LOGIC


Auto
Stop
Run
*
Hand
Stop
*
Auto
Net Ctrl
*
*
Net Ref
*
* - User Specific Logic


F O R
Drive Intelligent
Input Terminals
Jog Enable
Hand Auto
IO to Logic
Controller
Speed
Drive Intelligent
Input Terminals
0
100

Appendix A

The AF91DNA Communication Interface allows an AF91 Adjustable Frequency Drive to operate as a slave device on a DeviceNet network. The Communication Interface supports Explicit Messages and Polled I/O Messages of the predefined master/slave connection set. It *does not* support the Explicit Unconnected Message Manager (UCMM).

This appendix defines the DeviceNet message types, class services, and objects that are supported by the Communication Interface.

As a group 2 slave device, the Communication Interface supports the following message types.

CAN Identifier Field	Group 2 Message Type
10xxxxxx111	Duplicate MAC ID Check Messages
10xxxxxx110	Unconnected Explicit Request Messages
10xxxxxx101	Master I/O Poll Command Messages
10xxxxxx100	Master Explicit Request Messages
10xxxxxx011	Slave Explicit Response Messages
01111xxxxxx	Slave Poll Response Messages

xxxxxx = Communication Interface Node Address

The Communication Interface supports the following object classes.

Class Code		Object Name
1	01_{hex}	Identity
2	02_{hex}	Message Router
3	03_{hex}	DeviceNet
4	04_{hex}	Assembly
5	05_{hex}	Connection
160	$A0_{hex}$	AF91 Basic Parameters
161	$A1_{hex}$	AF91 I/O Signals
166	$A6_{hex}$	AF91 Protections
167	$A7_{hex}$	AF91 Trip History
168	$A8_{hex}$	AF91 Control Supervisor
169	$A9_{hex}$	AF91 AC Drive
170	AA_{hex}	AF91 Actual Data

The Services supported by these object classes are shown below.

Service Code		Service Name	Identity		Message Router		DeviceNet		Assembly		Connection		AF91 Objects	
			Class	Inst	Class	Inst	Class	Inst	Class	Inst	Class	Inst	Class	Inst
5	05_{hex}	Reset (Type 0, 1, 100, & 101)		Yes										
14	$0E_{hex}$	Get_Attribute_Single	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
16	10_{hex}	Set_Attribute_Single		Yes		Yes		Yes		Yes		Yes		Yes
20	14_{hex}	Error Response	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
24	18_{hex}	Get Member	Yes		Yes		Yes		Yes		Yes		Yes	
75	$4B_{hex}$	Allocate_Master/Slave_Connection_Set						Yes						
76	$4C_{hex}$	Release_Master/Slave_Connection_Set						Yes						
		Vendor Specific Services												
71	47_{hex}	Cutler-Hammer Get Member	Yes		Yes		Yes		Yes		Yes		Yes	

The attribute list that follows includes information on the Data Type of each attribute. The following tables explain the Data, Structure, and Array Type codes used in the Data Type column. For further information see Appendix J of the ODVA DeviceNet specification.

Elementary Data Types

Data Type Name	Data Type Code (in hex)	Data Type Description
BOOL	C1	Logical Boolean with values TRUE and FALSE
INT	C3	Signed 16–bit integer value
USINT	C6	Unsigned 8–bit integer value
UINT	C7	Unsigned 16–bit integer value
UDINT	C8	Unsigned 32–bit integer value
BYTE	D1	bit string - 8-bits
WORD	D2	bit string - 16-bits
SHORT_ STRING	DA	character sting (1 byte per character, 1 byte length indicator)

Constructed Data Types

Type Code	Description
A1	Abbreviated array type encoding
A2	Formal structure type encoding

In the following list, attributes shown in **bold** type (also designated EE in the Access Rule column) are stored in the non-volatile memory of the AF91DNA Communication Interface and maintain their value after a power loss. Attributes which are *italicized* are stored in the non-volatile memory of the AF91 Drive, after the EE Store Operation attribute (Class = 160 Instance = 1 Attribute ID = 15) has been set to 1 to initiate the store function, and maintain their value after a power loss. All other settable attributes will power up at their default values.

Attributes listed in the shaded area in the following list must be set during the configuration of the AF91DNA while it is connected to an AF91 inverter. These attributes must be stored in non-volatile memory to maintain their values after power loss.

The Description column for many of the AF91DNA attributes in this appendix contains “Actual Value” equations. Each equation provides the information needed to calculate the actual value of the number represented by its respective attribute in real world units. For example: The equation for the Output Frequency attribute in the AF91DNA Actual Data Object is “Actual value (Hz) = (Output Frequency)/100”. Substituting a value of 6000 for the Output Frequency in this equation yields an output frequency actual value of 60 Hz.

Before operating over DeviceNet, NetCtrl (Class = 168 Instance = 1 Attribute ID = 5) must be set to 1, “Network Control”. Setting the speed of the drive over DeviceNet requires that NetRef (Class = 169 Instance = 1 Attribute ID = 4) be set to 1, “Network Reference”.

Identity Object

CLASS CODE 1 (01_{HEX})

Identity Class 1 (01_{hex}) - Instance 1 (01_{hex})

Attr #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	Vendor	Get_Attribute_Single	C7	68, 68, 68	Identification of each vendor by number. 68 = Cutler-Hammer
2 02_{hex}	Device Type	Get_Attribute_Single	C7	12, 12, 12	Indication of the general type of product 12 = Communication Adapter
3 03_{hex}	Product Code	Get_Attribute_Single	C7	91, 91, 91	This is a code assigned by the vendor to describe the device. Product code determined by interrogating the connected drive. 91 = AF91
4 04_{hex}	Revision	Get_Attribute_Single	A2 02 C6 C6	{x,y}, {x,y}, {x,y}	Revision of the item the Identity Object represents x = Major revision y = Minor revision
5 05_{hex}	Status	Get_Attribute_Single	D2	N/A, 0, 4069	Summary Status of the Device. Defined in ODVA DeviceNet spec. Low byte: Bit 0 = Owned Bit 1 = 0 Bit 2 = Configured Bit 3 = 0 Bit 4 = 0 Bit 5 = User fault Bit 6 = Node fault Bit 7 = System fault
6 06_{hex}	Serial Number	Get_Attribute_Single	C8	N/A, 680787968, 805306367	Serial Number of the device
7 07_{hex}	Product Name	Get_Attribute_Single	DA	"AF91DNA", "AF91DNA", "AF91DNA"	Human readable identification.
8 08_{hex}	State	Get_Attribute_Single	C6	N/A, 0, 5	Present state of the device as represented by the state transition diagram. 0 = Nonexistent 1 = Device Self Testing 2 = Standby 3 = Operational 4 = Major Recoverable Fault 5 = Major Unrecoverable Fault
9 09_{hex}	Configuration Consistency Value	Get_Attribute_Single EE	C7	N/A, 0, 65535	Contents identify configuration of device
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance Maximum of 8 characters

DeviceNet Object

CLASS CODE 3 (03_{HEX})

DeviceNet Class 3 (03_{hex}) - Instance 1 (01_{hex})

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	MAC ID	Get_Attribute_Single	C6	N/A, 0, 63	Node Address. (Set by DIP switch)
2 02_{hex}	Baud Rate	Get_Attribute_Single	C6	N/A, 0, 2	The baud rate of the device. (Set by DIP switch) Values are: 00 = 125K 01 = 250K 02 = 500K 03 = Fault
5 05_{hex}	Allocation Information	Get_Attribute_Single	A2 02 D1 C6	N/A, N/A, N/A	Allocation Choice Master's Mac ID Struct of: BYTE: Allocation Choice byte Bit 0 = explicit messaging Bit 1 = Polled I/O USINT:Master's Mac ID 0-63 valid 255 = unallocated
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance Maximum of 8 characters

Assembly Object

CLASS CODE 4 (04_{HEX})

Assembly Class 4 (04_{hex}) Instance 100 (64_{hex}) Output-Speed

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
3 03_{hex}	Data	Get_Attribute_Single	A2 05 A1 01 C1 C6 C3	N/A, N/A, N/A	Output Assembly Data Basic Control + Speed Byte 0 (Bit/Byte n maps to: [Class][Instance][Attribute]) Bit 0:Run Fwd[A8][1][3] Bit 1:Run Rev[A8][1][4] Bit 2:Fault Reset[A8][1][C] Bit 5:NetCtrl[A8][1][5] Bit 6:NetRef[A9][1][4] Byte 1 Empty Byte 2 Speed Ref (LB)[A9][1][8] Byte 3 Speed Ref (HB)
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters

Output 100 (Default)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0		Net Ref	Net Ctrl			Fault Reset	Run Rev	Run Fwd
Byte 1								
Byte 2	Speed Reference – RPM (Low Byte)							
Byte 3	Speed Reference – RPM (High Byte)							

Assembly Class 4 (04_{hex}) Instance 101 (65_{hex}) Output Speed-Accel-Decel

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
3 03_{hex}	Data	Get_Attribute_Single	A2 07 A1 01 C1 C6 C3 C7 C7	N/A, N/A, N/A	Output Assembly Data Basic Control + Speed + Acceleration + Deceleration Byte 0 (Bit/Byte n maps to: [Class][Instance][Attribute]) Bit 0:Run Fwd[A8][1][3] Bit 1:Run Rev[A8][1][4] Bit 2:Fault Reset[A8][1][C] Bit 5:NetCtrl[A8][1][5] Bit 6:NetRef[A9][1][4] Byte 1 Empty Byte 2 Speed Ref (LB)[A9][1][8] Byte 3 Speed Ref (HB) Byte 4 Accel Time(LB)[A0][1][A] Byte 5 Accel Time(HB) Byte 6 Decel Time(LB)[A0][1][B] Byte 7 Decel Time(HB)
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance Maximum of 8 characters

Output 101

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0		Net Ref	Net Ctrl			Fault Reset	Run Rev	Run Fwd
Byte 1								
Byte 2	Speed Reference – RPM (Low Byte)							
Byte 3	Speed Reference – RPM (High Byte)							
Byte 4	Acceleration Time – Seconds x 10 (Low Byte)							
Byte 5	Acceleration Time – Seconds x 10 (High Byte)							
Byte 6	Deceleration Time – Seconds x 10 (Low Byte)							
Byte 7	Deceleration Time – Seconds x 10 (High Byte)							

Assembly Class 4 (04_{hex}) Instance 105 (69_{hex}) Input-Speed

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
3 03_{hex}	Data	Get_Attribute_Single	A2 05 A1 01 C1 D1 C3	N/A, N/A, N/A	Input Assembly Data Running Status + Input/Output + Actual Speed Byte 0 (Bit/Byte n maps to: [Class][Instance][Attribute]) Bit 2: Running Fwd [A8][1][7] Bit 3: Running Rev [A8][1][8] Byte 1[AA][1][6] Bit 0: Input 1 Status Bit 1: Input 2 Status Bit 2: Input 3 Status Bit 3: Input 4 Status Bit 4: Input 5 Status Bit 5: Output 11 Status Bit 6: Output 12 Status Bit 7: Output AL Status Byte 2 Speed Act (LB)[A9][1][7] Byte 3 Speed Act (HB)
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance Maximum of 8 characters

Input 105 (Default)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0					Running 2 (Reverse)	Running 1 (Forward)		
Byte 1	Output AL Status	Output 12 Status	Output 11 Status	Input 5 Status	Input 4 Status	Input 3 Status	Input 2 Status	Input 1 Status
Byte 2	Speed Actual – RPM (Low Byte)							
Byte 3	Speed Actual – RPM (High Byte)							

Assembly Class 4 (04_{hex}) Instance 106 ($6A_{hex}$) - Input-Speed-Net

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
3 03_{hex}	Data	Get_Attribute_Single	A2 05 A1 01 C1 D1 C3	N/A, N/A, N/A	Input Assembly Data Running Status + Net Data + Input/Output + Actual Speed Byte 0 (Bit/Byte n maps to: [Class][Instance][Attribute]) Bit 0: Faulted [A8][1][A] Bit 2: Running Fwd [A8][1][7] Bit 3: Running Rev [A8][1][8] Bit 5: CtrlFromNet [A8][1][F] Bit 6: RefFromNet [A9][1][1D] Bit 7: At Reference [A9][1][3] Byte 1 [AA][1][6] Bit 0: Input 1 Status Bit 1: Input 2 Status Bit 2: Input 3 Status Bit 3: Input 4 Status Bit 4: Input 5 Status Bit 5: Output 11 Status Bit 6: Output 12 Status Bit 7: Output AL Status Byte 2 Speed Act (LB) [A9][1][7] Byte 3 Speed Act (HB)
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance Maximum of 8 characters

Input 106

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	At Reference	Ref From Net	Ctrl From Net		Running 2 (Reverse)	Running 1 (Forward)		Faulted (Drive or I/F)
Byte 1	Output AL Status	Output 12 Status	Output 11 Status	Input 5 Status	Input 4 Status	Input 3 Status	Input 2 Status	Input 1 Status
Byte 2	Speed Actual – RPM (Low Byte)							
Byte 3	Speed Actual – RPM (High Byte)							

Note: “Faulted” indicates drive is faulted OR interface is faulted (User, Node, or System fault in status byte of Identity Object). Check Fault Source attribute for source of last fault.

Assembly Class 4 (04_{hex}) Instance 107 ($6B_{hex}$) - Input-Speed-Refer-Current

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
3 03_{hex}	Data	Get_Attribute_Single	A2 07 A1 01 C1 D1 C3 C3 C7	N/A, N/A, N/A	Input Assembly Data Running Status + Net Data + Input/Output + Actual Speed + Speed Ref + Output Current Byte 0 (Bit/Byte n maps to: [Class][Instance][Attribute]) Bit 0: Faulted [A8][1][A] Bit 2: Running Fwd [A8][1][7] Bit 3: Running Rev [A8][1][8] Bit 5: CtrlFromNet [A8][1][F] Bit 6: RefFromNet [A9][1][1D] Bit 7: At Reference [A9][1][3] Byte 1 [AA][1][6] Bit 0: Input 1 Status Bit 1: Input 2 Status Bit 2: Input 3 Status Bit 3: Input 4 Status Bit 4: Input 5 Status Bit 5: Output 11 Status Bit 6: Output 12 Status Bit 7: Output AL Status Byte 2 Speed Act (LB) [A9][1][7] Byte 3 Speed Act (HB) Byte 4 Speed Ref (LB) [A9][1][8] Byte 5 Speed Ref (HB) Byte 6 Out Current (LB) [AA][1][2] Byte 7 Out Current (HB)
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance Maximum of 8 characters

Input 107

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	At Reference	Ref From Net	Ctrl From Net		Running 2 (Reverse)	Running 1 (Forward)		Faulted (Drive or I/F)
Byte 1	Output AL Status	Output 12 Status	Output 11 Status	Input 5 Status	Input 4 Status	Input 3 Status	Input 2 Status	Input 1 Status
Byte 2	Speed Actual – RPM (Low Byte)							
Byte 3	Speed Actual – RPM (High Byte)							
Byte 4	Input Speed Reference – RPM (Low Byte)							
Byte 5	Input Speed Reference – RPM (High Byte)							
Byte 6	Output Current – amps x 100 (Low Byte)							
Byte 7	Output Current – amps x 100 (High Byte)							

Note: "Faulted" indicates drive is faulted OR interface is faulted (User, Node, or System fault in status byte of Identity Object). Check Fault Source attribute for source of last fault.
Input Speed Reference is the network reference value (CtrlFromNet = 1) or the analog reference value (CtrlFromNet = 0)

Assembly Class 4 (04_{hex}) Instance 108 ($6C_{hex}$) - Input-Speed-Refer-Current-PID

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
3 03_{hex}	Data	Get_Attribute_Single	A2 08 A1 01 C1 D1 C3 C3 C7 C8	N/A, N/A, N/A	Input Assembly Data Running Status + Net Data + Input/Output + Actual Speed + Speed Ref + Output Current + PID Feedback Data Byte 0 (Bit/Byte n maps to: [Class][Instance][Attribute]) Bit 0: Faulted [A8][1][A] Bit 2: Running Fwd [A8][1][7] Bit 3: Running Rev [A8][1][8] Bit 5: CtrlFromNet [A8][1][F] Bit 6: RefFromNet [A9][1][1D] Bit 7: At Reference [A9][1][3] Byte 1 [AA][1][6] Bit 0: Input 1 Status Bit 1: Input 2 Status Bit 2: Input 3 Status Bit 3: Input 4 Status Bit 4: Input 5 Status Bit 5: Output 11 Status Bit 6: Output 12 Status Bit 7: Output AL Status Byte 2 Speed Act (LB) [A9][1][7] Byte 3 Speed Act (HB) Byte 4 Speed Ref (LB) [A9][1][8] Byte 5 Speed Ref (HB) Byte 6 Out Current (LB) [AA][1][2] Byte 7 Out Current (HB) Byte 8 PID FB(LB) [AA][1][3] Byte 9 PID FB(MB) Byte 10 PID FB (MB) Byte 11 PID FB (HB)
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance Maximum of 8 characters

Input 108

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Byte 0	At Reference	Ref From Net	Ctrl From Net		Running 2 (Reverse)	Running 1 (Forward)		Faulted (Drive or I/F)	
Byte 1	Output AL Status	Output 12 Status	Output 11 Status	Input 5 Status	Input 4 Status	Input 3 Status	Input 2 Status	Input 1 Status	
Byte 2	Speed Actual – RPM (Low Byte)								
Byte 3	Speed Actual – RPM (High Byte)								
Byte 4	Input Speed Reference – RPM (Low Byte)								
Byte 5	Input Speed Reference – RPM (High Byte)								
Byte 6	Output Current – amps x 100 (Low Byte)								
Byte 7	Output Current – amps x 100 (High Byte)								
Byte 8	PID Feedback Data (Low Byte)								Refer to AF91 Instruction Manual, Section 8: PID Feedback Diagram
Byte 9	PID Feedback Data (Next Byte)								
Byte 10	PID Feedback Data (Next Byte)								
Byte 11	PID Feedback Data (High Byte)								

Note: "Faulted" indicates drive is faulted OR interface is faulted (User, Node, or System fault in status byte of Identity Object). Check Fault Source attribute for source of last fault.
Input Speed Reference is the network reference value (CtrlFromNet = 1) or the analog reference value (CtrlFromNet = 0)

DeviceNet Connection Object

CLASS CODE 5 (05_{HEX})

DeviceNet Connection Class 5 (05_{hex}) Instance 1 (01_{hex}) - Explicit Connection

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	State	Get_Attribute_Single	C6	3, 0, 5	State of the object
2 02_{hex}	Instance Type	Get_Attribute_Single	C6	0, 0, 0	Indicates either I/O or messaging connection 0=Explicit
3 03_{hex}	Transport Class Trigger	Get_Attribute_Single	C6	131, 131, 131	Defines Behavior of the connection
4 04_{hex}	Produced Cnxn Id	Get_Attribute_Single	C7	N/A, 0, 2047	Placed in CAN Identifier Field when the Connection Transmits
5 05_{hex}	Consumed Cnxn Id	Get_Attribute_Single	C7	N/A, 0, 2047	CAN Identifier Field value that denotes message to be received
6 06_{hex}	Initial Comm Characteristics	Get_Attribute_Single	C6	33, 33, 33	Defines the Message Group(s) across which productions and consumptions associated with this
7 07_{hex}	Produced Connection Size	Get_Attribute_Single	C7	516, 516, 516	Maximum number of bytes transmitted across this Connection
8 08_{hex}	Consumed Connection Size	Get_Attribute_Single	C7	516, 516, 516	Maximum number of bytes received across this Connection
9 09_{hex}	Expected Packet Rate	Get_Attribute_Single, Set_Attribute_Single	C7	2500, 0, 65535	Defines timing associated with this Connection
12 $0C_{hex}$	Watchdog Timeout Action	Get_Attribute_Single, Set_Attribute_Single	C6	1, 1, 3	Defines how to handle Inactivity/Watchdog timeouts. 1-Auto Delete 3-Defered Delete
13 $0D_{hex}$	Produced Connection Path Length	Get_Attribute_Single	C7	0, 0, 0	Number of bytes in the produced_connection_path length attribute
14 $0E_{hex}$	Produced Connection Path	Get_Attribute_Single	A1 01 C6	N/A, N/A, N/A	Application Obj. producing data on this connection
15 $0F_{hex}$	Consumed Connection Path Length	Get_Attribute_Single	C7	0, 0, 0	Number of bytes in the consumed_connection_path length attribute
16 10_{hex}	Consumed Connection Path	Get_Attribute_Single	A1 01 C6	N/A, N/A, N/A	Specifies the Application Object(s) that are to receive the data consumed by this Connection Object
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance Maximum of 8 characters

DeviceNet Connection Class 5 (05_{hex}) Instance 2 (02_{hex}) - Polled IO Connection

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	State	Get_Attribute_Single	C6	1, 0, 7	State of the object
2 02_{hex}	Instance Type	Get_Attribute_Single	C6	1, 1, 1	Indicates either IO or messaging connection 1 = I/O
3 03_{hex}	Transport Class Trigger	Get_Attribute_Single	C6	130, 130, 130	Defines Behavior of the connection
4 04_{hex}	Produced Cnxn Id	Get_Attribute_Single	C7	N/A, 0, 2047	Placed in CAN Identifier Field when the Connection Transmits
5 05_{hex}	Consumed Cnxn Id	Get_Attribute_Single	C7	N/A, 0, 2047	CAN Identifier Field value that denotes message to be received
6 06_{hex}	Initial Comm Characteristics	Get_Attribute_Single	C6	1, 1, 1	Defines the Message Group(s) across which productions and consumptions associated with this
7 07_{hex}	Produced Connection Size	Get_Attribute_Single	C7	4, 0, 12	Maximum number of bytes transmitted across this Connection
8 08_{hex}	Consumed Connection Size	Get_Attribute_Single	C7	4, 0, 8	Maximum number of bytes received across this Connection
9 09_{hex}	Expected Packet Rate	Get_Attribute_Single, Set_Attribute_Single	C7	0, 0, 65535	Defines timing associated with this Connection
12 $0C_{hex}$	Watchdog Timeout Action	Get_Attribute_Single, Set_Attribute_Single	C6	0, 0, 2	Defines how to handle Inactivity/Watchdog timeouts 0-Transition to time out 1-Auto Delete 2-Auto Reset
13 $0D_{hex}$	Produced Connection Path Length	Get_Attribute_Single	C7	6, 6, 6	Number of bytes in the produced_connection_path length attribute
14 $0E_{hex}$	Produced Connection Path	Get_Attribute_Single	A1 01 C6	N/A, N/A, N/A	Application Obj. producing data on this connection
15 $0F_{hex}$	Consumed Connection Path Length	Get_Attribute_Single	C7	6, 6, 6	Number of bytes in the consumed_connection_path length attribute
16 10_{hex}	Consumed Connection Path	Get_Attribute_Single	A1 01 C6	N/A, N/A, N/A	Specifies the Application Object(s) that are to receive the data consumed by this Connection Object
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters

AF91DNA Basic Parameters Object

CLASS CODE 160 ($A0_{HEX}$)

AF91DNA Basic Parameters Class 160 ($A0_{hex}$) - Instance 1 (01_{hex})

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	*Frequency Source Setting*	Get_Attribute_Single, Set_Attribute_Single	C6	*, 0, 2	**DO NOT USE.** Use NetRef (Class = 169 Instance = 1 Attribute ID = 4) and Local Reference Source (Class = 160 Instance = 1 Attribute ID = 12) instead. AF91 Function A01 Selects frequency source. 0=Potentiometer on front case of drive. 1=Terminal Block 2=Network DO NOT set during Run state Note: Setting of NetRef and Local Reference Source 160 1 12) will force the value of this attribute. Do not change the value of this attribute without an understanding of the Speed Reference Logic. Consult the factory for details. * Default value depends on model and date code of AF91 inverter.
2 02_{hex}	*Running Command Source Setting*	Get_Attribute_Single, Set_Attribute_Single	C6	*, 1, 2	**DO NOT USE.** Use NetCtrl instead (Class = 168 Instance = 1 Attribute = 5). AF91 Function A02 Selects START command source. 1=Terminal Block 2=Network DO NOT set during Run state Note: Setting of NetCtrl will force the value of this attribute * Default value depends on model and date code of AF91 inverter.
3 03_{hex}	*Base Frequency*	Get_Attribute_Single, Set_Attribute_Single	C7	60, 50, 360	AF91 Function A03 Sets the frequency at which inverter outputs rated voltage Actual value (Hz) = Base Frequency CANNOT be set during Run state. Does not require EE Store Operation to save attribute in nonvolatile drive memory
4 04_{hex}	*Maximum Frequency*	Get_Attribute_Single, Set_Attribute_Single	C7	60, 50, 360	AF91 Function A04 Sets the maximum inverter output frequency. During transient operation the output frequency may exceed the maximum by up to 1.5 Hz. Actual value (Hz) = Maximum Frequency CANNOT be set during Run state. Does not require EE Store Operation to save attribute in nonvolatile drive memory
5 05_{hex}	*Output Frequency Reference*	Get_Attribute_Single, Set_Attribute_Single	C7	0, 0, 36000	**DO NOT USE.** Use SpeedRef instead (Class = 169 Instance = 1 Attribute = 8). AF91 Function A20 Sets the desired output frequency. Actual Value (Hz) = (Output Frequency Reference)/100
6 06_{hex}	*Torque Boost Adjustment*	Get_Attribute_Single, Set_Attribute_Single	C6	11, 0, 99	AF91 Function A42 Increases voltage at low speeds to increase starting torque. CAUTION High values may burn out motor or cause inverter trips. Actual value (volts) = Torque Boost Adjustment
7 07_{hex}	*Torque Boost Frequency*	Get_Attribute_Single, Set_Attribute_Single	C7	100, 0, 500	AF91 Function A43 Sets upper limit of torque boost in percent of base frequency. Actual value (%) = (Torque Boost Frequency)/10
8 08_{hex}	*V Gain Adjustment*	Get_Attribute_Single, Set_Attribute_Single	C6	100, 50, 100	AF91 Function A45 Sets the voltage gain of the V/F characteristic. Actual value (%) = V Gain Adjustment It is advisable that this attribute be set while the drive is stopped. Large step changes in the value of this attribute may cause inverter trips.

9 09_{hex}	*Carrier Frequency Setting*	Get_Attribute_Single, Set_Attribute_Single	C6	50, 5, 160	AF91 Function b83 Sets the carrier frequency of the inverter's PWM output. Actual value (kHz) = (Carrier Frequency Setting)/10 CANNOT be set during Run state.
10 $0A_{hex}$	*Acceleration Time Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	100, 1, 30000	AF91 Function F02 Sets the time to ramp up from 0 to Maximum Frequency. Ramping to an Output Frequency less than the Maximum Frequency will occur in a proportionally shorter time. Actual value (sec) = (Acceleration Time Setting)/10
11 $0B_{hex}$	*Deceleration Time Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	100, 1, 30000	AF91 Function F03 Sets the time to ramp down from Maximum Frequency to 0. Ramping from an Output Frequency less than the Maximum Frequency will occur in a proportionally shorter time. Actual value (sec) = (Deceleration Time Setting)/10
12 $0C_{hex}$	**Local Reference Source**	Get_Attribute_Single, Set_Attribute_Single EE	C1	0, 0, 1	Specifies the source of the reference when NetRef transitions to zero. 0=KeyPad (or POT) 1=Terminal Block
13 $0D_{hex}$	**Number of Poles**	Get_Attribute_Single, Set_Attribute_Single EE	C6	4, 2, 255	Specifies the number of poles in the connected motor. Used in speed calculations. Using data from an AC motor's nameplate: Motor Poles = (120*Hz)/RPM (rounded up to the nearest whole number) CANNOT be set during run state
14 $0E_{hex}$	*Motor Voltage*	Get_Attribute_Single, Set_Attribute_Single	C7	*, 200, 460	AF91 Function A82 Enter the nameplate voltage of the connected motor. Acceptable ranges: * 200 VAC Class: 200-240 (230 default) 400 VAC Class: 380-460 (460 default) Actual value (volts) = Motor Voltage CANNOT be set during Run state. Does not require EE Store Operation to save attribute in nonvolatile drive memory
15 $0F_{hex}$	EE Store Operation	Get_Attribute_Single, Set_Attribute_Single	C1	0. 0, 1	Stores the AF91 attributes in non-volatile memory of the inverter. Set = 1 Starts the attribute store process Get = 1 Store process not yet complete Get = 0 Store process complete
16 10_{hex}	**Polled Input Assembly Type**	Get_Attribute_Single, Set_Attribute_Single	C7	105, 105, 108	The input assembly used by the polled connection
17 11_{hex}	**Polled Output Assembly Type**	Get_Attribute_Single, Set_Attribute_Single	C7	100, 100, 101	The output assembly used by the polled connection
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters.

AF91DNA I/O Signals Object

CLASS CODE 161 ($A1_{HEX}$)

AF91DNA I/O Signals Class 161 ($A1_{hex}$) - Instance 1 (01_{hex})

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	*Selection of PID Control*	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	AF91 Function A71 Selects if PID control is operational. 0 = PID disabled 1 = PID enabled CANNOT be set during run state
2 02_{hex}	*P (Proportional) Gain Setting*	Get_Attribute_Single, Set_Attribute_Single	C6	10, 2, 50	AF91 Function A72 Sets the proportional gain of the PID control . Actual value = P Gain Setting/10
3 03_{hex}	*I (Integral) Gain Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	10, 0, 1500	AF91 Function A73 Sets the integral gain of the PID control. Actual value (1/sec) = (I Gain Setting)/10
4 04_{hex}	*D (Derivative) Gain Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	0, 0, 1000	AF91 Function A74 Sets the derivative gain of the PID control. Actual value (sec) =(D Gain Setting) /10
5 05_{hex}	*Scale Conversion of PID Control Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	100, 1, 9999	AF91 Function A75 (Scale Conversion of PID Control Setting) Scales the Target value equivalent to the feedback value. Refer to the PID feedback diagram in Section 8 of the AF91 Instruction Manual. Actual value = (Scale Conversion of PID Control Setting)/100 CANNOT be set during run state
6 06_{hex}	*Feedback Source Setting*	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	AF91 Function A76 Selects the source of the PID feedback. 0=OI terminal (current) 1=O terminal (voltage) CANNOT be set during run state
7 07_{hex}	*Output Frequency Display Scaling*	Get_Attribute_Single, Set_Attribute_Single	C7	10, 1, 999	AF91 Function b86 Sets the scaling of the d07 monitor. Actual value = (Output Frequency Display Scaling)/10 d07 display = actual value * output frequency
8 08_{hex}	*Level of Overload Signal Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	260, 0, 65535	AF91 Function C41 Sets the threshold of the overload signal. Used in conjunction with the intelligent output terminals. Range is 0 to 200% of rated inverter current. Actual value (amps) = (Level of Overload Signal Setting)/100 CANNOT be set during run state
9 09_{hex}	*Level of Deviation Signal Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	30, 0, 1000	AF91 Function C44 Sets the threshold of the deviation signal. The deviation is between the Target and the Feedback of the PID control. Used in conjunction with the intelligent output terminals. Actual value (% of full scale) = (Level of Deviation Signal Setting)/10 CANNOT be set during run state
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters.

AF91DNA Protections Object

CLASS CODE 166 ($A6_{HEX}$)

AF91DNA Protections Class 166 ($A6_{hex}$) - Instance 1 (01_{hex})

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	*Selection of DC Braking Operation*	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	AF91 Function A51 Selects if DC Braking is operational. 0=Disabled 1=Enabled CANNOT be set during run state
2 02_{hex}	*DC Braking Frequency Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	50, 0, 1000	AF91 Function A52 Sets the frequency at which DC Braking is enabled. Actual value (Hz) = (DC Braking Frequency Setting)/100 CANNOT be set during run state
3 03_{hex}	*DC Braking Waiting Time Setting*	Get_Attribute_Single, Set_Attribute_Single	C6	0, 0, 50	AF91 Function A53 Sets the time the drive free runs before DC Braking is turned on. Actual value (sec) = (DC Braking Waiting Time Setting)/100 CANNOT be set during run state
4 04_{hex}	*DC Braking Force Setting*	Get_Attribute_Single, Set_Attribute_Single	C6	0, 0, 100	AF91 Function A54 Sets the force of DC Braking. (100% is approximately the inverter's output current rating.) Actual value (%) = DC Braking Force Setting CANNOT be set during run state
5 05_{hex}	*DC Braking Time Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	0, 0, 600	AF91 Function A55 Sets the length of time DC Braking will operate. Actual value (sec) = (DC Braking Time Setting)/10 CANNOT be set during run state
6 06_{hex}	*Jump Frequency 1 Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	0, 0, 36000	AF91 Function A63 Sets frequency to "jump over" to avoid resonance with the load. Actual value (Hz) = (Jump Frequency 1 Setting)/100 CANNOT be set during run state
7 07_{hex}	*Jump Frequency Width 1 Setting*	Get_Attribute_Single, Set_Attribute_Single	C6	5, 0, 100	AF91 Function A64 Sets the width of the frequency jump. Actual value (Hz) = (Jump Frequency Width 1 Setting)/10 CANNOT be set during run state
8 08_{hex}	*Level of Electronic Thermal Setting*	Get_Attribute_Single, Set_Attribute_Single	C7	N/A, N/A, N/A	AF91 Function b12 Sets the level of electronic thermal protection in amps. Range is 50% to 120% of rated inverter current. Actual value (amps) = (Level of Electronic Thermal Setting)/100 CANNOT be set during run state. To avoid an E08 trip after changing this attribute, you must first do an EE Store (Class = 160 Instance = 1 Attribute ID = 15).
9 09_{hex}	*Overload Restriction Level Setting*	Get_Attribute_Single, Set_Attribute_Single	C6	*, 20, 200	AF91 Function b22 Sets the overload limit. Range is 20 to 200% of rated inverter current. Actual value (%) = Overload Restriction Level Setting CANNOT be set during run state * Default value depends on model and date code of AF91 inverter.
10 $0A_{hex}$	*Selection of Operation at FRS Signal Canceled*	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	AF91 Function b88 (Selection of Operation at FRS Signal Canceled) Sets the restart mode after Free Run Stop is cancelled. 0=Restart from 0Hz 1=Rotating restart CANNOT be set during run state

11 $0B_{hex}$	**Safe State Behavior**	Get_Attribute_Single, Set_Attribute_Single EE	C6	0, 0, 2	Selects Safe State response to errors which specify safe state operation. Currently only a loss of connection other than by de-allocation is a safe state error. **Warning**: Review the application for safe operation before specifying a value for this attribute. 0 = Stop & Fault 1 = Hold Last Speed 2 = Preset Speed/Direction
12 $0C_{hex}$	**Safe State Preset Frequency**	Get_Attribute_Single, Set_Attribute_Single EE	C7	0, 0, 36000	Sets safe state frequency reference if the Safe State Behavior attribute specifies "Preset Speed/Direction" and RefFromNet = 1. **Warning**: Review the application for safe operation before specifying a value for this attribute. Inverter will require external stop. Actual value (Hz) = (Safe State Preset Frequency)/100
13 $0D_{hex}$	**Safe State Preset Direction**	Get_Attribute_Single, Set_Attribute_Single EE	C6	2, 0, 3	Sets safe state direction of rotation if the Safe State Behavior attribute specifies "Preset Speed/Direction" and CtrlFromNet = 1. **Warning**: Review the application for safe operation before specifying a value for this attribute. Inverter will require external stop. 0 = Forward 1 = Reverse 2 = Hold Last Direction 3 = Go To Opposite Direction
14 $0E_{hex}$	**TB5-Comm Loss**	Get_Attribute_Single, Set_Attribute_Single EE	C6	2, 0, 2	Assigns FRS (Free Run Stop) or Ext Trip to AF91 input terminal 5 and sets polarity of terminal 5 to normally closed. 1=FRS assigned 2=Ext Trip assigned 0=Hold Last Speed. Uses customer assignment of input 5 or CH factory default if unassigned. Values 1 & 2 override any other assignment for input terminal 5.
15 $0F_{hex}$	**Idle State Behavior**	Get_Attribute_Single, Set_Attribute_Single EE	C6	0, 0, 1	Selects response to Receive_Idle event when running. **Warning**: Review the application for safe operation before specifying a value for this attribute 0 = Stop 1 = Hold Last Speed
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters.

AF91DNA Trip History Object

CLASS CODE 167 ($A7_{HEX}$)

AF91DNA Trip History Class 167 ($A7_{hex}$) - Instance 1 (01_{hex})

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	Clear Trip History	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	Clear Trip History 0-1 = Trip History Clear 0= No Action
2 02_{hex}	Trip Counter	Get_Attribute_Single	C6	N/A, 0, 255	Refer to AF91 Instruction Manual
3 03_{hex}	Accumulated Time At Trip 1	Get_Attribute_Single	C8	N/A, 0, 3054198	Time of latest trip Accumulated hours since trip history was last cleared via the Keypad. Actual value (hours) = (Accumulated Time At Trip 1)/10 Refer to AF91 Instruction Manual
4 04_{hex}	Cause of Trip 1	Get_Attribute_Single	C6	N/A, 0, 255	Trip code for latest trip. See Table of Error Codes in AF91DNA Instruction Leaflet.
5 05_{hex}	Output Frequency at Trip 1	Get_Attribute_Single	C7	N/A, 0, 36000	Inverter output frequency at the time of Trip 1. Actual value (Hz) = (Output Frequency at Trip 1)/100
6 06_{hex}	Output Current at Trip 1	Get_Attribute_Single	C7	N/A, 0, 65535	Inverter output current at the time of Trip 1. Actual value (%) = (Output Current at Trip 1)/100
7 07_{hex}	Accumulated Time At Trip 2	Get_Attribute_Single	C8	N/A, 0, 3054198	Time of 2nd latest trip Accumulated hours since trip history was last cleared via the Keypad. Actual value (hours) = (Accumulated Time At Trip 2)/10 Refer to AF91 Instruction Manual
8 08_{hex}	Cause of Trip 2	Get_Attribute_Single	C6	N/A, 0, 255	Trip code for 2nd latest trip. See Table of Error Codes in AF91DNA Instruction Leaflet.
9 09_{hex}	Output Frequency At Trip 2	Get_Attribute_Single	C7	N/A, 0, 36000	Inverter output frequency at the time of Trip 2. Actual value (Hz) = (Output Frequency at Trip 2)/100
10 $0A_{hex}$	Output Current At Trip 2	Get_Attribute_Single	C7	N/A, 0, 65535	Inverter output current at the time of Trip 2. Actual value (%) = (Output Current at Trip 2)/100
11 $0B_{hex}$	Accumulated Time at Trip 3	Get_Attribute_Single	C8	N/A, 0, 3054198	Time of 3rd latest trip Accumulated hours since trip history was last cleared via the Keypad. Actual value (hours) = (Accumulated Time At Trip 3)/10 Refer to AF91 Instruction Manual
12 $0C_{hex}$	Cause of Trip 3	Get_Attribute_Single	C6	N/A, 0, 255	Trip code for 3rd latest trip. See Table of Error Codes in AF91DNA Instruction Leaflet.
13 $0D_{hex}$	Output Frequency at Trip 3	Get_Attribute_Single	C7	N/A, 0, 36000	Inverter output frequency at the time of Trip 3. Actual value (Hz) = (Output Frequency at Trip 3)/100
14 $0E_{hex}$	Output Current at Trip 3	Get_Attribute_Single	C7	N/A, 0, 65535	Inverter output current at the time of Trip 3. Actual value (%) = (Output Current at Trip 3)/100
15 $0F_{hex}$	*Trip Code 1*	Get_Attribute_Single	DA	N/A, “E 01”, “E 35”	AF91 inverter trip code for latest trip, Trip 1. See Table of Error Codes in AF91DNA Instruction Leaflet or Section 9 of the AF91 Inverter Instruction Manual.
16 10_{hex}	*Trip Code 2*	Get_Attribute_Single	DA	N/A, “E 01”, “E 35”	AF91 inverter trip code for 2nd latest trip, Trip 2. See Table of Error Codes in AF91DNA Instruction Leaflet or Section 9 of the AF91 Inverter Instruction Manual.

17 11_{hex}	*Trip Code 3*	Get_Attribute_Single	DA	N/A, "E 01", "E 35"	AF91 inverter trip code for 3rd latest trip, Trip 3. See Table of Error Codes in AF91DNA Instruction Leaflet or Section 9 of the AF91 Inverter Instruction Manual.
18 12_{hex}	Fault Source	Get_Attribute_Single	C6	0, 0, 255	Identifies the cause of the latest fault. Bit 0=Drive Trip Bit 5=User Bit 6=Node Bit 7=System Bit 5-7 are vendor specific fault bits from the Identity Object status byte.
19 13_{hex}	**Event List 1**	Get_Attribute_Single	C6	N/A, 0, 33	The most recent recorded event in node. See Table of Af91DNA Interface Error Codes for more detail.
20 14_{hex}	**Event List 2**	Get_Attribute_Single	C6	N/A, 0, 33	The 2nd most recent recorded event in node. See Table of Af91DNA Interface Error Codes for more detail.
21 15_{hex}	**Event List 3**	Get_Attribute_Single	C6	N/A, 0, 33	The 3rd most recent recorded event in node. See Table of Af91DNA Interface Error Codes for more detail.
22 16_{hex}	**Event List 4**	Get_Attribute_Single	C6	N/A, 0, 33	The 4th most recent recorded event in node. See Table of Af91DNA Interface Error Codes for more detail.
23 17_{hex}	**Event Count**	Get_Attribute_Single	C7	N/A, 0, 65535	Number of events logged in Event List since "out of box". Value will wrap around to 0 after 65535.
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters.

AF91DNA Control Supervisor Object

CLASS CODE 168 ($A8_{HEX}$)

AF91DNA Control Supervisor Class 168 ($A8_{hex}$) - Instance 1 (01_{hex})

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
3 03_{hex}	Run1	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	Run Forward Request 0 = Do Not Request Run Forward (Stop if both Run1 and Run2 = 0) 1 = Request Run Forward
4 04_{hex}	Run2	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	Run Reverse Request 0 = Do Not Request Run Reverse (Stop if both Run1 and Run2 = 0) 1 = Request Run Reverse
5 05_{hex}	NetCtrl	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	Specifies source of Run/Stop control. 1→0 = Local Control (AF91 terminal) 0→1= Request for control from the network Note: Setting the value of NetCtrl will force the value of Running Command Source Setting (Class = 160 Instance = 1 Attribute ID = 2)
7 07_{hex}	Running1	Get_Attribute_Single	C1	N/A, 0, 1	Feedback indicating whether drive is running forward 0=Not Running Forward 1=Running Forward.
8 08_{hex}	Running2	Get_Attribute_Single	C1	N/A, 0, 1	Feedback indicating whether drive is running reverse 0 = Not Running Reverse 1 = Running Reverse
10 $0A_{hex}$	Faulted	Get_Attribute_Single	C1	N/A, 0, 1	Indicates an AF91DNA fault or an AF91 trip has occurred. See also Fault Source attribute. 0= Not faulted 1= Faulted
12 $0C_{hex}$	FaultRst	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	Sends trip reset command to AF91. 0→1 = Fault Reset 0 = No Action
15 $0F_{hex}$	CtrlFromNet	Get_Attribute_Single	C1	N/A, 0, 1	Actual Status of Run/Stop Control Source 0=Control is Local (Terminal) 1=Control is from Network (Response to NetCtrl “request”)
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters.

AF91DNA AC Drive Object

CLASS CODE 169 ($A9_{HEX}$)

AF91DNA AC Drive Class 169 ($A9_{hex}$) - Instance 1 (01_{hex})

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
3 03_{hex}	AtReference	Get_Attribute_Single	C1	N/A, 0, 1	Indicates when drive's operating frequency is at the frequency reference set point (-0.5 to +1.5 Hz). 0=Not at reference 1=At reference
4 04_{hex}	NetRef	Get_Attribute_Single, Set_Attribute_Single	C1	0, 0, 1	Specifies source of Frequency Reference. 1→0=Reference is Local (See also Local Reference Source attribute) 0→1=Request for reference control form the network Note: Setting the value of NetRef will force the value of Frequency Source Setting (Class = 160 Instance = 1 Attribute ID = 1)
7 07_{hex}	Speed Actual	Get_Attribute_Single	C7	N/A, 0, 21600	The Actual RPM of the Motor (calculated). Actual value (RPM) = Speed Actual
8 08_{hex}	SpeedRef	Get_Attribute_Single, Set_Attribute_Single	C7	0, 0, 21600	Speed Reference in RPM units (AF91 Function A20) Actual value (RPM) = SpeedRef Requires RefFromNet = 1 to set. Setting this attribute causes a corresponding change in Output Frequency Reference.
29 $1D_{hex}$	RefFromNet	Get_Attribute_Single	C1	N/A, 0, 1	Actual status of speed reference source. 0=Speed reference is local. See also Local Reference Source attribute. 1=Speed reference is from network (Response to NetRef "request")
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters.

AF91DNA Actual Data Object

CLASS CODE 170 (AA_{HEX})

AF91DNA Actual Data Class 170 (AA_{hex}) - Instance Attributes 1 (01_{hex})

ATT #	Attribute Name	Access Rule (Services)	Data Type	Values: Default, Minimum, Maximum	Description
1 01_{hex}	Output Frequency	Get_Attribute_Single	C7	N/A, 0, 36000	AF91 Function d01 Inverter output frequency. Actual value (Hz) = (Output Frequency)/100
2 02_{hex}	Output Current	Get_Attribute_Single	C7	N/A, 0, 65535	AF91 Function d02 Inverter output current Actual value (amps) = (Output Current)/100
3 03_{hex}	PID Feedback Data	Get_Attribute_Single	C8	N/A, 0, 999900	AF91 Function d04 Representation of the feedback used by the PID control. See the PID feedback diagram in Section 8 of the AF91 Instruction Manual. Actual value = (PID Feedback Data)/100
4 04_{hex}	Scaled Conversion Output Frequency	Get_Attribute_Single	C8	N/A, 0, 3596400	AF91 Function d07 Actual value = (Scaled Conversion Output Frequency)/100 Output frequency is scaled by the Output Frequency Display Scaling attribute
5 05_{hex}	Output Current Monitor (%)	Get_Attribute_Single	C7	N/A, 0, 2000	Output current displayed in percent of inverter rated current. Actual value (%) = (Output Current Monitor)/10
6 06_{hex}	Digital IO	Get_Attribute_Single	C6	N/A, 0, 255	Digital Signals on AF91 intelligent terminals Bit 0: Input 1 Status Bit 1: Input 2 Status Bit 2: Input 3 Status Bit 3: Input 4 Status Bit 4: Input 5 Status Bit 5: Output 11 Status Bit 6: Output 12 Status Bit 7: Output AL Status
176 $B0_{hex}$	**User Label**	Get_Attribute_Single, Set_Attribute_Single EE	DA	N/A, N/A, N/A	User Supplied name for the Instance. Maximum of 8 characters.