ASTRA

Allen Bradley DF1 Device Driver

www.renuelectronics.com

⌘ Tables of Contents ⌘

⌘ Preface ⌘

This document introduces user to the ALLEN BRADLEY PLC DF1 Port Device Driver. It contains technical information about DF1 Device Driver. This document gives you a broad idea of how to use DF1 Device Driver with Astra.

The whole document persists you information about the capabilities and technical details of DF1 Device Driver and how to use the driver.

⌘ Introduction ⌘

The intent of this document is to assist users of the ALLEN BRADLEY DF1-SERIES Driver in conjunction with the Astra MMI software package. A general knowledge of ALLEN BRADLEY DF1-SERIES PLC is assumed. The addressing scheme of the ALLEN BRADLEY programming software with some slight modification is explained in the subsequent chapters. Description of the different data types and the addressing scheme should be understood before attempting to use the driver in a Astra project.

The driver is intended to be used only on one PC serial port at a time and connected to only one PLC (as the DF1 is a Point to Point protocol). The driver will operate upto 19200 baud but 9600 baud is recommended. At 19200 baud, more errors are encountered in the communication and the end result is often a slower total throughput than running at 9600 baud.

The optimization features described in this document can improve performance, but they are not essential for use.

⌘ Technical & Communication Details ⌘

PLC Make :	Allen Bradley
PLC Modles :	SLC5/03, SLC5/04, Micrologix.
Communication Protocol :	DF-1.
Communication Parameters :	Baud Rate - 9600 Parity - None Data Bits - 8 Stop Bits - 1 Error Detection - CRC
Cable Connections :	Use a standard RS-232C cable configuration.
Node ID :	The Node ID specified in ASTRA should match with that defined in the PLC.

⌘ Data Types and Addressing ⌘

Data Types:

The following is a description of how the DF1 driver interprets the information from the PLC as different data types. The PLC programmer is responsible for ensuring that the referenced locations can logically be interpreted as correct type. This is particularly important for floating point numbers, as there are such bit configurations, that are incompatible with the IEEE floating point format.

All 16 bit word and 32 bit double words must start on a 16 bit boundary. It is possible to overlap double words using this format. For example N7:0 and N7:1 both defined as data type Long, would share the 16 bit word at location N7:1 as either their low word or high word respectively. Since this is probably not desirable behavior, care should be taken to avoid overlap situations.

⌘ Data Types and Addressing ⌘

Sr. No.	Data Type	Driver Information Format
1.	Discrete	A single bit.
2.	Unsigned Integer	Unsigned 16 bit word Bit 0 is the LSB (Least Significant Bit) Bit 15 is the MSB (Most Significant Bit)
3.	Integer	Signed 16 bit word Bit 15 is the sign bit Bit 0 is the LSB Bit 14 is the MSB
4.	Long	Signed 32 bit word Bits 0-15 are the high word Bits 16-31 are the low word Bit 15 is the sign bit Bit 14 is the MSB Bit 16 is the LSB
5.	Real	Thirty-two bit IEEE floating point number Bit 31 is the sign bit Bits 23-30 are the exponent Bit 30 is MSB Bit 23 is LSB Bits 0-22 are the mantissa

Note: An exponent of 0x00 or 0xff cannot be interpreted correctly as an IEEE floating point number. The driver will return a value of 0.0 for these numbers even though this is not an accurate value.

Addressing:

Note : Since user can configure I/O modules in any slots of the rack and frame contains fixed file number (0 for outputs and 1 for inputs) it is important that he should understand memory maps for I/O modules. They are all sequential for both input and output modules. For example, if user has put two 16 channel digital input modules at slot numbers 3 and 12 and also one analog input modules at slot 4 (having four 16 bit registers) address mapping will be I1/0 to I1/15 for slot 3, I1/16 to I1/79 for slot 4 and I1/80 to I1/95 for slot 12 respectively.

⌘ Data Types and Addressing ⌘

Outputs

Type - Discrete
Format - O<File No>/<Bit> For SLC500
Format - O0/<Bit> For Micrologix

Range SLC500 :- File No. 0
Bit 0-992

Micrologix :- File No. 0
Bit 0-31

Examples:
For SLC500

O0/0 The first bit of the output module in file 0
O0/8 The 9th bit of the output module in file 0

For Micrologix
O0/8 The 9th bit of output module of Micrologix

Inputs

Type - Discrete
Format - I<File No>/<Bit> For SLC500
Format - I1/<Bit> For Micrologix

Range SLC500 :- File No. 1
Bit 0-992

Micrologix :- File No. 1
Bit 0-31

Examples:
For SLC500
I1/0 The first bit of the input module in file 1
I1/14 The 15th bit of the input module in file 1

For Micrologix
I1/0 The first bit of input module of Micrologix
I1/14 The 15th bit of input module of Micrologix

Binary

Type - Discrete

Format - B<File No>/<Bit> For SLC500 PLC
Format - B3/<Bit> For Micrologix

Range SLC500 :- File: 3, 10-255
Bit: 0-255

Micrologix :- File: 3
Bit: 0-255

Examples:
For SLC500

B3/0 The first bit of binary file 3
B10/0 The first bit of binary file 10
B3/255 The last bit of binary file 3

For Micrologix
B3/0 The first bit of binary file of Micrologix
B3/15 The 16th and last bit of the first word of binary file of Micrologix

⌘ Data Types and Addressing ⌘

Type - Unsigned integer, Integer
Format - B3:<Word>
Range - Word: 0-31

Examples:
B3:0 The first word of binary file 3
B3:31 The last (32nd) word of binary file 3

Type - Long, Real
Format - B3:<DoubleWord>
Range - DoubleWord: 0-30

Examples:
B3:0 The first double word of binary file 3
B3:30 The last double word of binary file 3

Timers

Type - Long, Real
Format - T<File>:<Timer><Word Suffix>
For SLC500 PLC
Format - T4:<Timer>.<Word Suffix>
For Micrologix PLC

Range - File: 4, 10-255 For SLC500 PLC
Timer: 0-255
Word Suffix: A Accumulator
P Preset
Timer: 0-39 For Micrologix PLC
Word Sufix: A Accumulator
P Preset

Examples:
For SLC500
T4:0A The accumulator register of the first timer in timer file 4
T9:39P The preset register of the 40th timer in timer file 9

For Micrologix
T4:0.A The accumulator register of the first timer
T4:39.P The preset register of the 40th timer

Counters

Type - Unsigned Integer, Integer
Format - C<File>:<Counter>
<Word Suffix>For SLC500 PLC
Format - C5:<Counter>.<Word Suffix>
For Micrologix
Range - File: 5, 10-255 For SLC500 PLC
Counter: 0-255
Word Suffix: A Accumulator
P Preset
Counter: 0-31 For Micrologix PLC
Word Suffix: A Accumulator
P Preset

Examples:

For SLC500
C5:0A The accumulator register of the first timer counter in counter file 5
C13:39P The preset register of the 40th counter in counter file 13

For Micrologix
C5:0.A The accumulator register of the first counter
C5:39.P The preset register of the 40th counter in counter file 5

Control Registers

Type - Unsigned integer, Integer, Long, Real
Format - R<File>:<Word>
For SLC500 PLC
Format - R6:<Word> For Micrologix PLC

⌘ Data Types and Addressing ⌘

Range - File: 6, 10-255 For SLC500 PLC
Word: 0-255
Word: 0-15 For Micrologix PLC

Examples:

For SLC500 R10:4 5th control register in control file 10

For Micrologix R6:4 5th control register

Examples:

For SLC500

N7:0 The first register in long/real file 7
N9:103 The last register in long/real file 9

For Micrologix

N7:0 The first register in long/real file 7
N7:103 The last register in long/real file 7

Integer Registers

Type - Unsigned integer, Integer
Format - N<File>:<Word>For SLC500 PLC
Format - N7:<Word> For Micrologix

Range - File: 7, 10-255 For SLC500 PLC
Word: 0-255
Range Word:0-104 For Micrologix

Examples:

For SLC500

N7:0 The first register in integer file 7
N59:255 The last (256th) register in integer file 59

For Micrologix

N7:0 The first register in integer file
N7:104 The last (105th) register in integer file

Type - Long, Real
Format - N<File>:<Word> For SLC500
Format - N7:<Word> For Micrologix

Range - File: 7, 10-255 For SLC500 PLC
Word: 0-254

Range Word: 0-103 For Micrologix

Floating Point Registers

Type - Long integer, Real
Format - F<File>:<Word>For SLC500
Format - NA For Micrologix

Range - File: 8-254 For SLC500
Word: 0-254
Range NA For Micrologix

Examples:

For SLC500

F8:0 The first register in floating point file 8

F242:254 The last register in floating point file 242

⌘ Data Types and Addressing ⌘

Address Ranges (FOR SLC 5/03 & SLC 5/04) :

Memory Reference Types	Selector / Segment	Low Address (FileNo)	High Address (FileNo)	Low Address	High Address	Represe-ntation	Read /Write	Data Types
Output Registers	O	000	000	000	992	Dec	R	Dis
Input Registers	I	001	001	000	992	Dec	R	Dis
Bit Registers	B	003	254	000	4095	Dec	R/W	Dis
Bit Registers	B	003	254	000	031	Dec	R/W	UInt,Int, Long,Real
Timers (Accumulator)	T	004	254	000.A	039.A	Dec	R	Long, Real
Timers (Preset)	T	004	254	000.P	039.P	Dec	R/W	Long, Real
Counters (Accumulator)	C	005	254	000.A	031.A	Dec	R	Long, Real
Counters (Preset)	C	005	254	000.P	031.P	Dec	R/W	Long, Real
Control Registers	R	006	254	000	015	Dec	R/W	Uint,Int, Long,Real
Integer Registers	N	007	254	000	104	Dec	R/W	UInt,Int, Long,Real
Floating point	F	008	254	000	254	Dec	R/W	Long, Real

⌘ Data Types and Addressing ⌘

Example :-

Input	:- **I1/12**	i.e. the 13th input bit of the input module in file 1.
Output	:- **O0/0**	i.e. the 1st output bit of the output module in file 0.
Bit	:- **B10/0**	i.e. the 1st binary bit of the binary file 10.
Timer	:- **T4:0A**	i.e. the accumulator register of the 1st timer in timer file 4.
Counter	:- **C5:0P**	i.e. the preset register of the 1st counter in counter file 5.
Control	:- **R6:4**	i.e. the 5th control register in control file 6.
Integer	:- **N7:0**	i.e. the integer register zero and one reference as double word in integer file 7
Float	:- **F8:0**	i.e. the 1st floating point register of the floating point file 8.

Address Ranges (FOR Micrologix) :

Memory Reference Types	Selector / Segment	Low Address (FileNo)	High Address (FileNo)	Low Address	High Address	Represe-ntation	Read /Write	Data Types
Output Registers	O	000	000	000	015	Dec	R	Dis
Input Registers	I	001	001	000	031	Dec	R	Dis
Bit Registers	B	003	003	000	511	Dec	R/W	Dis
Bit Registers	B	003	003	000	031	Dec	R/W	UInt,Int, Long,Real
Timers (Accumulator)	T	004	004	000.A	039.A	Dec	R	Long, Real
Timers (Preset)	T	004	004	000.P	039.P	Dec	R/W	Long, Real
Counters (Accumulator)	C	005	005	000.A	031.A	Dec	R	Long, Real

⌘ Data Types and Addressing ⌘

Memory Reference Types	Selector / Segment	Low Address (FileNo)	High Address (FileNo)	Low Address	High Address	Represe-ntation	Read /Write	Data Types
Counters (Preset)	C	005	005	000.P	031.P	Dec	R/W	Long, Real
Control Registers	R	006	006	000	015	Dec	R/W	UInt,Int, Long,Real
Integer Registers	N	007	007	000	104	Dec	R/W	UInt,Int, Long,Real

Example :-

Input :- **I1/0** the first input bit of Input File
Output :- **O0/0** the first output bit of Input File
Bit :- **B3:31** the thirty-second word of the Binary File
Timer :- **T4:0.A** the accumulator register of the first timer
T4:39.P the preset register of the fortieth timer
Counter :- **C5:0.A** the accumulator register of the first counter
C5:39.P the preset register of the fortieth counter
Control :- **R6:4** the length field of the fifth control register
Integer :- **N7:0** the first integer register word in integer file 7

⌘ Optimizations ⌘

Use the following guidelines so that you can get an optimum performance from the driver PLC combination.

- Whenever possible use consecutive addresses, this reduces the overhead on the communication per requested data byte, word or double word.
- When a same address is to be used for two different tags in Astra, make sure that the scan time is the same for both the tags, this ensures that the address is fetched only once for both the tags.
- Use higher scan rates whenever the application allows to do so, this ensures that the critical tags with lower scan rates are fetched with minimum overhead.

⌘ Errors ⌘

The entire time an Astra project is running, the Event Logger displays the status and any errors that the program generates. The driver utilizes the Event Logger to display error messages regarding the driver. Below are the error messages, the probable cause and most likely solution to all the errors that the driver can generate. Error nos. are displayed in the event logger in case of the error situation

Errors displayed as strings

1. NULL Pointer for Login Data
2. NULL Pointer for Project Path
3. NULL Pointer for Tag Table
4. NULL Handle for Data Manager

Explanation : Internal Fatal Error.
Action : Contact Astra support.

5. Insufficient Memory for Request Manager
6. Insufficient Memory for Transaction Manager
7. Insufficient Memory for Device Manager

Explanation : Internal Fatal Error.
Action : Try making more memory available for the project.

8. Cannot Pagelock Tag Table
Explanation : Internal Fatal Error.
Action : Contact Astra support.

9. Cannot Open File PLCTAG.DAT 10. Cannot Read File PLCTAG.DAT
Explanation : Internal Fatal Error. The input file PLCTAG.DAT does not exist or is corrupt.
Action : Open the project in the configuration mode and close it, this process recompiles the PLCTAG.DAT file.

11. Insufficient Memory for Tag 12. Insufficient Memory for Tag2
13. Insufficient Memory for Tag Container 14. Insufficient Memory for Node
15. Insufficient Memory for Node Container
Explanation : Internal Fatal Error.
Action : Try making more memory available for the project.

16. No Tags in the Project
Explanation : Internal Fatal Error. The driver detected no valid tags in the project.
Action : Recheck the project in the configuration mode. See if any tags are assigned to this particular device.
See if the Node details are correct.

17. No Valid Nodes in the Project
Explanation : Internal Fatal Error. The driver detected no valid nodes in the project.
Action : Recheck the project in the configuration mode. See if the Node details are correct.

18. Multidrop not Supported
Explanation : Internal Fatal Error. An attempt was made to attach two nodes on the same driver when Multidrop is not supported.
Action : Recheck the project in the configuration mode. See if the Node details are correct.

19. Multiple nodes with same ID

Explanation : Internal Fatal Error. An attempt was made to attach two nodes on the same driver with same Node IDs.

Action : Recheck the project in the configuration mode. See if the Node details are correct.

20. Insufficient Memory for Request
21. Insufficient Memory for Request2
22. Insufficient Memory for Request Container
23. Insufficient Memory for Dummy Request
24. Insufficient Memory for Action
25. Insufficient Memory for Action Container

19. Cannot Create Communication Window

Explanation : Internal Fatal Error.

Action : Try making more memory available for the project.

20. Cannot Open Communication Port

Explanation : Internal Fatal Error. Could not initialize the Communication port for the given settings.

Action : For the selected Communication port, check for –

- ⊙ If the port physically exists.
- ⊙ If the Communication hardware uses standard base addresses. COM1 uses hex 3F8 and COM2 uses hex 2F8.
- ⊙ If there is any IRQ contention at the hardware level. COM1 uses IRQ4 and COM2 uses IRQ3.
- ⊙ If any other program is already using the Communication port you have requested for.
- ⊙ If any DOS level TSRs are running which are using the Communication port you have requested for.
- ⊙ If a mouse driver is installed on the same Communication port you have requested for in Windows environment.
- ⊙ If a mouse driver is installed on the same Communication port you have requested for on DOS environment.
- ⊙ If you have directly manipulated the PROJECT.INI file section [COM1] or [COM2], check if the settings for Baud Rate, Data Bits, Stop Bits and the Parity are standard. Try using the Communication port setting utility provided with Astra in case you are in doubts about the standard settings.

21. Cannot Build Communication DCB

Explanation : Internal Fatal Error. Could not initialize the Communication port for the given settings.

Action : If you have directly manipulated the PROJECT.INI file section [COM1] or [COM2], check if the settings for Baud Rate, Data Bits, Stop Bits and the Parity are standard. Try using the Communication port setting utility provided with Astra in case you are in doubts about the standard settings.

22. Cannot Set Communication State

Explanation : Internal Fatal Error. Could not initialize the Communication port for the given settings.

Action : If you have directly manipulated the PROJECT.INI file section [COM1] or [COM2], check if the settings for Baud Rate, Data Bits, Stop Bits and the Parity are standard. Try using the Communication port setting utility provided with Astra in case you are in doubts about the standard settings.

23. NULL Pointer for Model Names

Explanation : Internal Fatal Error.

Action : Contact Astra support.

24. Read Queue Full

25. Device Time Out

Explanation : The Device did not respond and the Device driver timed out. The Driver will retry the request to Device for a specified number of times and if the Device still does not respond the driver will HALT its transactions with the Device.

Action : If this happens during **initialization**, check –

- ⊙ Whether the Device power is on.
- ⊙ Whether the cable connections to the device are proper.
- ⊙ Whether the Node ID settings are proper in case the Device supports it.
- ⊙ Whether the Device model is the same as configured in the Node Configuration.
- ⊙ Whether the Communication hardware is proper and works.
- ⊙ Whether strong EMI or RFI fields are existent which cause noise on the Communication line.
- ⊙ Whether some turnaround delay is required, try changing the entries in the DRIVERS.INI file. This may be typically required for faster PCs on which Astra runs.

If this happens during the **Run**, check –

- ⊙ Whether other applications block the Windows, in such a case the retry mechanism will normally re-establish the Communication.
- ⊙ Whether the cable connections have been disturbed.
- ⊙ Whether the Device has malfunctioned.
- ⊙ Whether the Communication hardware is proper and works.

26. Invalid IEEE Format

Explanation : The 32 bits read from the Device contained bit values such that it could not be interpreted as a valid IEEE format.

Action : Use OEM software and initialize floating type tags in the plc.

27. Write Queue Full

Explanation : The write request sent by the Astra is queued for faster execution, the current limit for the queue size is 300. If the queue is full this message will be prompted and the latest request will be ignored.

Action : Go to the project configuration file and put an entry with section name "QueueSize". Under this section name, put a key name "WriteQueue", so that it looks like:

[QueueSize]
WriteQueue=WXY
Where,
WXY can be upto 5000.

28. This error number is no more in use !!!

29. No Valid Tags in the Project

Explanation : Internal Fatal Error. The driver detected no valid tags in the project.

Action : Recheck the project in the configuration mode. See if any tags are assigned to this particular device. See if the Node details are correct.

30. Insufficient Memory for Register 31. Insufficient Memory for Tag Container2
32. Insufficient Memory for Register Container
Explanation : Internal Fatal Error.
Action : Try making more memory available for the project.

33. Tag Address Invalid 34. Tag Address Invalid2
Explanation : The address entered for a Tag is invalid.
Action : Reconfigure the project and check.

35. Driver Scan Halted 36. Driver Scan Halted2
Explanation : The driver has stopped communicating with the device. This may happen in two situations –
- ⊙ When the initial scan is complete - in this case this is just a status information.
- ⊙ When time-out has occurred and retry for establishing communication has failed.

Action : In the second case check –
- ⊙ If the cable connections have been disturbed.
- ⊙ If the Device has malfunctioned.
- ⊙ If the Communication hardware is proper and works.

37. Cannot Find INI File Entry, Setting Default Port
Explanation : The [PROTOCOL] section in PROJECT.INI does not have the driver name against the COM1 or the COM2 entry. In such a case default COM1 is selected as the Communication port.
Action : Run the Communication port setting utility provided with Astra and set all the parameters properly.

38. This error number is no more in use!!! 39. Cannot Run without initialization
40. Cannot Run without initialization2 41. Cannot Write without initialization
42. Cannot Build frames without initialization
Explanation : Due to some abnormal termination in a previous run the Device Driver has not unloaded itself and hence could not deinitialize itself.
Action : Unload Windows and restart again.

43. NULL Pointer for Queue
Explanation : Internal Fatal Error.
Action : Contact Astra support.

44. Invalid IEEE Format2
Explanation : This might be caused due to –
- ⊙ Presence of junk values at a particular memory area of your PLC.
- ⊙ Accessing a particular type of tag with improper data type, say accessing a floating point data type as a word.

Action : Doing the following actions in case of above mentioned causes, should help
- ⊙ Using your OEM software, you can fill in proper values at places which have junk values.
- ⊙ Use proper data types for your tags.

45. Cannot Page lock Buffer
Explanation : Internal Fatal Error.
Action : Contact Astra support.

46. Device Response Delay
Explanation : The Device did not respond and the Device driver timed out. The Driver will retry the request to Device for a specified number of times and if the Device still does not respond the driver will HALT its transactions with the Device.
Action : If this happens during initialization check –
- ⊙ If the Device power is on.
- ⊙ If the cable connections to the device are proper.
- ⊙ If the Device model is the same as configured in the Node Configuration.
- ⊙ If the Communication hardware is proper and works.
- ⊙ If strong EMI or RFI fields are existent which cause noise on the Communication line.

If this happens during the Run check –
- ⊙ If in case other applications block the Windows, in such a case the retry mechanism will normally re-establish the Communication.
- ⊙ If the cable connections have been disturbed.
- ⊙ If the Device has malfunctioned.
- ⊙ If the Communication hardware is proper and works.

47. Response Check Sum Error

Explanation : The Device did respond but the bytes received were corrupt. The Driver will retry the request to Device.

Action : If this happens during **initialization** check –

- ⊙ If the Communication hardware is proper and works.
- ⊙ If strong EMI or RFI fields are existent which cause noise on the Communication line.
- ⊙ If the Communication port settings are proper.

If this happens during the **Run** check –

- ⊙ If the cable connections have been disturbed.
- ⊙ If the Device has malfunctioned.
- ⊙ If the Communication hardware is proper and works.

48. Data Over Flow

Explanation : Unexpected data in large volume was received on the Communication port.

Action : Check –

- ⊙ If the cable connections have been disturbed.
- ⊙ If the Device has malfunctioned.
- ⊙ If the Communication hardware is proper and works.

49. Model Name Invalid

Explanation : Internal Fatal Error. The model name associated with a particular Node was invalid.

Action : Open the project in the configuration mode check the model in the Node Configuration.

50. Cannot Open File PLCTAG.DAT 2 51. Cannot Read File PLCTAG.DAT 2
52. Cannot Read File PLCTAG.DAT 3

Explanation : Internal Fatal Error. The input file PLCTAG.DAT does not exist or is corrupt.

Action : Open the project in the configuration mode and close it, this process recompiles the PLCTAG.DAT file.

53. This error number is no more in use !!! 54.This error number is no more in use !!!

55. Invalid number for conversion from BCD to WORD

Explanation : The 16 bits given for write from Astra to the Device contained bit values such that it could not be interpreted as a valid BCD format. Write will not be done in these cases.

Action : Avoid such values.

56. Invalid number for conversion from BCD to DWORD

Explanation : The 32 bits given for write from Astra to the Device contained bit values such that it could not be interpreted as a valid BCD format. Write will not be done in these cases.

Action : Avoid such values.

57. Invalid number for conversion from BCD to WORD

Explanation : The 16 bits given for write from Astra to the Device contained bit values such that it could not be interpreted as a valid BCD format. Write will not be done in these cases.

Action : Avoid such values.

58. Invalid number for conversion from BCD to DWORD

Explanation : The 32 bits given for write from Astra to the Device contained bit values such that it could not be interpreted as a valid BCD format. Write will not be done in these cases.

Action : Avoid such values.

59. Error Composing Write Request

Explanation : Write request could not be composed. This may happen in two cases –

- ⊙ Invalid number for write.
- ⊙ Write Queue full.

Action : Avoid non interpretable values, Avoid writing too fast .

60. Error Composing Read Request After Write

Explanation : A read request immediately following a write request could not be composed. This may happen in two cases –

- ⊙ Invalid number for write.
- ⊙ Write Queue full.

Action : Follow the following actions –

- ⊙ Avoid non interpretable values.
- ⊙ Kindly refer to error number 34 to increase the size of the write queue.

61. Node Failed.

Explanation : Internal Fatal Error. The Node was not able to communicate. In case of Multidrop PLC system the node ID given to the nodes may be same or cable from PC to PLC may be faulty.

Action : Open the project in the configuration mode check the Node Configuration and close it. For Multidrop communication check the node ID. Check the cable.

62. Cannot Open File NODES.DAT 63.Cannot Read File NODES.DAT 2

Explanation : Internal Fatal Error. The input file NODES.DAT does not exist or is corrupt.

Action : Open the project in the configuration mode and close it, this process recompiles the NODES.DAT file.

64. Node set on by user.

Explanation : Not an error . It indicates that node is selected by the user. For Astra generated default tags for a PLC, in that if command tag is 0 then this message is displayed.

Action : None , as it indicates that node is selected by the user.

65. Node set off by user.

Explanation : Not an error . It indicates that node is unselected by the user. For Astra generated default tags for a PLC in that if command tag is 1 then this message is displayed.

Action : None, as it indicates that node is unselected by the user.

66. Node manager procedure address not defined.
Explanation : Internal Fatal Error.
Action : Contact Astra support.

310. Error(Garbage).
Explanation : Garbage in data frame – this will most likely occur when there is noise in the communication line. Also, kindly refer to error number 54.
Action : Please refer to error number 54.

311. Error(NAK).
Explanation : Response from PLC had not acknowledged – this may happen when the tag is not configured for access, or, is physically non-existent. Event logger will show 311.
Action : If the tag is not already configured, please configure with OEM software.

Renu Electronics Pvt Ltd.
S.No. 2/6, Baner Road,
Pune 411045, India.
Tel: + 91 20 2729 2840,
Fax: + 91 20 2729 2839
Email: info@renuelectronics.com
Website: www.renuelectronics.com

TÜV CERT EN ISO 9001 : 2000 Certificate : 0410001296 RWTÜV