

# ASTRA

Modbus ASCII Device Driver

[www.renuelectronics.com](http://www.renuelectronics.com)

## ⌘ Tables of Contents ⌘

Preface	3
1. Introduction	3
2. Technical and Communication Details	4
3. Data Types and Addressing	5
4. Optimisations	11
5. Errors	11

## ⌘ Preface ⌘

This document introduces user to the [Modbus](#) ASCII Protocol Device Driver. It contains technical information about [Modbus](#) ASCII PLC Device Driver. This gives you a broad idea of how to use [Modbus](#) ASCII PLC Device Driver with Astra.

This document broadly, tells you about the capabilities and technical details of [Modbus](#) ASCII PLC Device Driver and how to use the driver.

## ⌘ Introduction ⌘

The intent of this document is to assist users of the [Modbus](#) ASCII Protocols PLC Driver in conjunction with the Astra MMI software package. A general knowledge of the [Modbus](#) ASCII Protocols PLC family is assumed. The addressing scheme of the 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 optimization features described in this document can improve performance, but they are not essential for use.

## ⌘ Technical & Communication Details ⌘

PLC Make :	Arromat
PLC Modles :	The driver supports 184/384,484,584,884, M84,984 , GENCON , COMMUNICATOR, CHAMELEON-6. PLC models and Generic for supporting other PLC's using MODBUS ASCII protocol.
PLC Memory :	WORD Memory.
Communication Protocol :	Modbus ASCII (Multi Point, Half Duplex)
Communication Parameters :	
<b>Baud Rate -</b>	9600
<b>Parity -</b>	EVEN
<b>Data Bits -</b>	7
<b>Stop Bits -</b>	1
Cable Connections :	The Modbus ASCII driver uses the RS232C standard for serial communication.
Node ID :	Use a node ID from 1 to 247.



**Note:** It is assumed that all the PLC models terminate the message with <CR> <LF>, without which time out will occur when used with ASTRA.

## ⌘ Data Types and Addressing ⌘

### Data Types:

The following is a description of how the Modbus 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. Some devices such as GENCON and COMMUNICATOR uses swapped floating-point format. In case of these models the WORD 's between Double WORD are swapped.

All 16 bit word and 32 bit double words must start on a 8 bit boundary for the Coil Status, Input status, Holding Register, Input Register. It should start on a 16 bit boundary for the Holding Register, Input Register. Specific to Modbus ASCII , it is possible to overlap double words using this format. Say, that the Double Words – 30 and 31, both are defined as data type long, so they would share the 16 bit location 30, as either their high word or low word respectively. Since, this is probably not the desirable behavior, care should be taken to avoid such overlapping situations.

### Address Ranges and Maximum Q/R Parameters (CPU – 184U/384U) :

Memory Reference Types	Low Add.	High Add.	Representation	Read /Write	Data Types
Output Coil	000001	100800	Dec	R/W	Dis
Input Coil	000001	100800	Dec	R	Dis
Input Register	300001	300100	Dec	R	UInt,Int, Long,Real
Holding Registers	400001	400100	Dec	R/W	UInt,Int, Long,Real

## ⌘ Data Types and Addressing ⌘

### Address Ranges and /maximum Q/R Parameters (CPU – 484U) :

Memory Reference Types	Low Add.	High Add.	Representation	Read /Write	Data Types
Output Coil	000001	000512	Dec	R/W	Dis
Input Coil	100001	100512	Dec	R	Dis
Input Register	300001	300254	Dec	R	UInt,Int, Long,Real
Holding Registers	400001	400032	Dec	R/W	UInt,Int, Long,Real

### Address Ranges and Maximum Q/R Parameters (CPU – 584U) :

Memory Reference Types	Low Add.	High Add.	Representation	Read /Write	Data Types
Output Coil	000000	002000	Dec	R/W	Dis
Input Coil	100000	102000	Dec	R	Dis
Input Register	300000	300125	Dec	R	UInt,Int, Long,Real
Holding Registers	400000	400125	Dec	R/W	UInt,Int, Long,Real

## ⌘ Data Types and Addressing ⌘

### Address Ranges and Maximum Q/R Parameters (CPU – 884U) :

Memory Reference Types	Low Add.	High Add.	Representation	Read /Write	Data Types
Output Coil	000000	002000	Dec	R/W	Dis
Input Coil	100000	102000	Dec	R	Dis
Input Register	300000	300125	Dec	R	UInt,Int, Long,Real
Holding Registers	400000	400125	Dec	R/W	UInt,Int, Long,Real

### Address Ranges and Maximum Q/R Parameters (CPU – M84U) :

Memory Reference Types	Low Add.	High Add.	Representation	Read /Write	Data Types
Output Coil	000001	000064	Dec	R/W	Dis
Input Coil	100001	100064	Dec	R	Dis
Input Register	300001	300032	Dec	R	UInt,Int, Long,Real
Holding Registers	400000	400004	Dec	R/W	UInt,Int, Long,Real

## ⌘ Data Types and Addressing ⌘

### Address Ranges and Maximum Q/R Parameters (CPU – 984U) :

Memory Reference Types	Low Add.	High Add.	Representation	Read /Write	Data Types
Output Coil	000001	002000	Dec	R/W	Dis
Input Coil	100001	102000	Dec	R	Dis
Input Register	300001	300125	Dec	R	UInt,Int, Long,Real
Holding Registers	400001	400125	Dec	R/W	UInt,Int, Long,Real

### Address Ranges for GENCON :

Memory Reference Types	Starting Address	End Address	Representation	Read /Write	Data Types
Discrete Inputs	100001	100061	Dec	R	Dis
Input Registers	300001	300041	Dec	R	UInt,Int, Long,Real

## ⌘ Data Types and Addressing ⌘

### Address Ranges for COMMUNICATOR:

Memory Reference Types	Start Add.	End Add.	Representation	Read /Write	Data Types
Coils	000001	000014	Dec	R/W	Dis
Discrete Inputs	100001	100016	Dec	R	Dis
Holding Registers	400001	400017	Dec	R/W	UInt,Int, Long,Real

### Address Ranges for CHAMELEON-6:

Memory Reference Types	Start Add.	End Add.	Representation	Read /Write	Data Types
Digital Values	000000	400009	Dec	R/W	Dis
Analog Values	400000	400010	Dec	R	UInt,Int, Real
Analog Values	400011	400220	Dec	R/W	UInt,Int, Real

#### About Double Words:

Double Word is a Astra term. It is not a PLC supported memory range. It is implemented by handling two consecutive words at a time.

#### About Selectors and Address:

100001 → 1 is Selector No.  
 00001 is Location No.

## ⌘ Data Types and Addressing ⌘

The Bit Interpretation is as follows:

ASTRA Tag Type	Size in Bits	Interpretation method in ASTRA	Example	
			Read Bits from PLC	Val.
Discrete	1	Discrete	1	1
Unsigned Integer	16	Decimal	00000000 00001011	11
Integer	16	Decimal	00000000 00010001	17
Large Integer	32	Decimal	00000000 00000000 00001011 11111111	307 1
Real	32	IEEE	01000000 00000000 00000000 00000000	2
			11000000 00000000 00000000 00000000	-2
			01000000 10000000 00000000 00000000	4
			01000000 11000000 00000000 00000000	6
			01000000 00100000 00000000 00000000	2.5
			01000000 00000000 00000000 00000000	



**Note** :In case Real numbers for GENCON and COMMUNICATOR the WORD 's ( 16 bits ) are swapped . For CHAMELEON-6 the floats are 2 bytes wide and 0-65535 is scaled to 0-100.0. Due to this user will get accuracy up to two decimal points only. There may be error in 1000's position after decimal point. No IEEE format is used for CHAMELEON-6.



## 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 a 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 the driver can generate.

### 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.



## Errors



9. Cannot Open 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.

10. Cannot Read File PLCTAG.DAT

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

26. Cannot Create Communication Window

**Explanation** : Internal Fatal Error.

**Action** : Try making more memory available for the project.



## Errors



### 27. 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.

### 28. 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.

### 29. 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.



## Errors



Try using the Communication port setting utility provided with Astra in case you are in doubts about the standard settings.

30. NULL Pointer for Model Names

31. Read Queue Full

**Explanation :** Internal Fatal Error.

**Action :** Contact Astra support.

32. 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.



## Errors



### 33. 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.

### 34. 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.

### 35. This error number is no more in use !!!

### 36. 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.

### 37. Insufficient Memory for Register

### 38. Insufficient Memory for Tag Container2

### 39. Insufficient Memory for Register Container

**Explanation :** Internal Fatal Error.

**Action :** Try making more memory available for the project.

### 40. Tag Address Invalid

### 41. Tag Address Invalid2.

**Explanation :** The address entered for a Tag is invalid.

**Action :** Reconfigure the project and check.

### 42. Driver Scan Halted

### 43. 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.



## Errors



- 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.

### 44. 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.

### 45. Cannot Initialise Driver Twice

**Explanation** : Due to some abnormal termination in a previous run the Device Driver has not unloaded itself and hence could not reinitialise itself.

**Action** : Restart the project.

### 46. Cannot Run Without Initialisation

### 47. Cannot Run Without Initialization2

### 48. Cannot Write Without Initialisation

### 49. Cannot Build Frames Without Initialisation

**Explanation** : Due to some abnormal termination in a previous run, the Device Driver has not unloaded itself and hence could not reinitialise itself.

**Action** : Restart the project.

### 50. NULL Pointer for Queue

**Explanation** : Internal Fatal Error.

**Action** : Contact ASTRA support.

### 51. Invalid IEEE Format2

**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 initialise floating type tags in the device.

### 52. Cannot Pagelock Buffer

**Explanation** : Internal Fatal Error.

**Action** : Contact ASTRA support.



## Errors



### 53. 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 **Initialisation** check –

- ⊙ If the Device is powered 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 working.
- ⊙ 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 working.

### 54. 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 **Initialisation** check –

- ⊙ If the Communication hardware is proper and working.
- ⊙ 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 working.



## Errors



### 55. 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 working.

### 56. 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 and close it.

### 57. Cannot Open File PLCTAG.DAT 2

### 58. Cannot Read File PLCTAG.DAT 2

### 59. 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.

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

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

62. Invalid number for conversion to BCD for WORD

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

**Action** : Use OEM software and initialise respective tags in the device.

63. Invalid number for conversion to BCD for 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.

64. Invalid number for conversion to BCD for WORD

**Explanation** : The 16 bits given for write from ProSoft/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.



## Errors



### 65. Invalid number for conversion to BCD for 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.

### 66. 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. Increase WriteQueueSize in project.ini file section [QueueSize]. Lower limit for WriteQueueSize is 300 and the upper limit is 5000.

### 67. 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 :** Avoid non interpretable values, Avoid writing too fast. Increase WriteQueueSize in project.ini file section [QueueSize]. Lower limit for WriteQueueSize is 300 and the upper limit is 5000.

### 68. Node Failed.

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

### 69. Cannot Open File NODES.DAT

### 70. 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.



## Errors



71. 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 DEVICE, in that if command tag is 0 then this message is displayed.

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

72. 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 DEVICE in that if command tag is 1 then this message is displayed.

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

73. Node manager proc address not defined.

**Explanation :** Internal Fatal Error.

**Action :** Contact ASTRA support.

74. The function code received in the query is not an allowable action for the slave , Function Code not supported by device.

**Explanation :** The user has tried to perform an action which is not allowed by the device. e.g. If the device does not support WRITE\_MULTIPLE\_REGISTERS function and user has tried to write then this error will occur. To elaborate if the model is defined as GENERIC and TAG is float/real , which is 4 bytes or 2 registers. And the device connected is CHAM6 . Now CHAM6 does not support simultaneous write on two registers . In such case this error will be reported by the device.

**Action :** Verify the project and PLC model. Ensure that the device model and Tag defined are supported by the device.

75. The data received in the query is not an allowable address for the slave , Invalid address for the device

**Explanation :** The request send by the user for the tag is with invalid address. The address specified by the user is an invalid address for the device. This will happen if the Tags are not defined properly or different Model is used.

**Action :** Ensure that correct model is used and the tag giving the error is defined properly.

76. A value contained in the query data field is not allowable value for the slave , Invalid value for the data.

**Explanation :** The request send by the user for the tag is with invalid data. The data specified by the user is an invalid data for the device. This will happen if the data entered is in invalid



## Errors



format. Normally this will happen in case of floating point data

**Action** : Ensure that correct model is used and the tag giving the error is defined properly.

77. An unrecoverable error occurred while the slave was attempting to perform the action ,  
Exception code 4 from device

**Explanation** : There was error while performing the action requested

78. The slave has accepted the request and is processing it

**Explanation** : Response from the device is going to take some time.

**Action** : Wait till response comes from the device.

79. Slave is engaged in processing a long-duration program command .

**Explanation** : The request is received by the device and is engaged in long duration processing  
and is unable to process the requested query.

**Action** : The same query should be sent after some time interval.



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](mailto:info@renuelectronics.com)

Website: [www.renuelectronics.com](http://www.renuelectronics.com)

