

# ASTRA

Telemecanique Programming Port Tsx47/ Tsx67 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	10
5. Limitations	10
6. Errors	11

## ⌘ Preface ⌘

This document introduces you to the Programming Port Protocol of Telemecanique [Tsx47/Tsx67](#) PLC device driver Version 32.0. It contains technical information about [Tsx47/Tsx67](#) device driver. This document gives you a broad idea of how to use [Tsx47/Tsx67](#) device driver with Astra.

This document broadly, tell you about the capabilities and technical details of [Tsx47/Tsx67](#) device driver and how to use the driver.

## ⌘ Introduction ⌘

The intent of this document is to assist users of the Programming Port Protocol of Telemecanique [Tsx47/Tsx67](#) PLC driver in conjunction with the MMI software package. A general knowledge of the Programming Port Protocol of Telemecanique [Tsx47/Tsx67](#) PLC family is assumed. The addressing scheme is that of the Telemecanique programming software with some slight modifications which are explained. 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 :	_____
PLC Modles :	The driver supports Tsx47, Tsx67
PLC Memory :	Word Memory.
Communication Protocol :	Telemecanique Tsx47 ( Point to Point, Half Duplex ).
Communication Parameters :	
<b>Baud Rate</b> -	9600
<b>Parity</b> -	ODD
<b>Data Bits</b> -	8
<b>Stop Bits</b> -	1
Cable Connections :	The Tsx47/Tsx67 driver runs on the RS232 standard for serial communication.
Node ID :	_____

## ⌘ Data Types and Addressing ⌘

### Data Types :

The following is a description of how the Telemecanique TSX07/TSX37 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 for the Internal bits, System bits. It should start on a 16 bit boundary for the System Words, Internal Words, Constant Words, Timers and Counters. Specific to TSX07/TSX37, it is possible to overlap double words using this format. Say, that the Internal Words - %MW00 and %MW01, both are defined as data type long, so they would share the 16 bit word at location %MW01, as either their high word or low word respectively. Since, this is probably not desirable behaviour, care should be taken to avoid overlap situations.

### I. Address Range :

#### Model Tsx47

Memory Range	Low Address	High Address	Read /Write	Data Types
Internal Coils	%B0	%B4095	RW	D
System Coils	%SY0	%SY63	R	D
System Words	%SW0	%SW63	R	UI,I
Internal Words	%W0	%W32767	RW	UI,I
Double Words	%DW0	%DW32766	RW	L,R
Input Bits	%I00,0	%I97,F	R	D
Output Bits	%O00,0	%O97,F	RW	D
Input Words	%I00	%I97	R	UI,I
Output Words	%O00	%O97	RW	UI,I

## ⌘ Data Types and Addressing ⌘

Memory Range	Low Address	High Address	Read /Write	Data Types
Timer Value	%T0,V	%T127,V	R	UI,I
Timer Preset	%T0,P	%T127,P	RW	UI,I
Counter Value	%C0,V	%C255,V	R	UI,I
Counter Preset	%C0,P	%C255,P	RW	UI,I
Monostable Value	%M0,V	%M127,V	R	UI,I
Monostable Preset	%M0,P	%M127,P	RW	UI,I

### \*\* **About Double Words :**

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

### \* **About Input/Output RACKS & SLOTS :**

I11,x → 1 is Rack No. ( Each Rack has 8 slots.)  
 1 is Slot No..  
 x is Bit position ( bet. 0 to F )



**Note:** RackNo. & SlotNo. should be represented in decimal format.

### **Model Tsx67**

Memory Range	Low Address	High Address	Read /Write	Data Types
Internal Coils	%B0	%B4095	RW	D
System Coils	%SY0	%SY63	R	D
System Words	%SW0	%SW63	R	UI,I
Internal Words	%W0	%W32767	RW	UI,I
Monostable Value	%DW0	%DW32766	RW	L,R

## ⌘ Data Types and Addressing ⌘

Memory Range	Low Address	High Address	Read /Write	Data Types
Input Bits	%I00,0	%IF7,F	R	D
Output Bits	%O00,0	%OF7,F	RW	D
Input Words	%I00	%IF7	R	UI,I
Output Words	%O00	%OF7	RW	UI,I
Timer Value	%T0,V	%T127,V	R	UI,I
Timer Preset	%T0,P	%T127,P	RW	UI,I
Counter Value	%C0,V	%C255,V	R	UI,I
Counter Preset	%C0,P	%C255,P	RW	UI,I
Monostable Value	%M0,V	%M127,V	R	UI,I
Monostable Preset	%M0,P	%M127,P	RW	UI,I

### \*\* **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 Input/Output RACKS & SLOTS :**

I11,x →

1 is Rack No. ( Each Rack has 8 slots.)

1 is Slot No..

x is Bit position ( bet. 0 to F )



**Note:** RackNo. & SlotNo. should be represented in decimal format.

## ⌘ Data Types and Addressing ⌘

The address usage is as follows :

Tag Type	Example	Actual Addresses fetched
Discrete	%B0 %SY0 %I00,F %O00,F	%B0, %SY0,%I00,F%,O00, F (where F represent 16th bit of that location )
Unsigned Integer and Integer	%SW0, %W0, %I00, %O00, %T0,V, %T0,P, %C0,V, %C0,P, %M0,V, %M0,P,	Where all Selector represent 0 <sup>th</sup> word  For Integer type, 15 <sup>th</sup> bit represent the sign bit
Large Integer	%DW000	%W000 - Low Word %W001 - High Word
Real	%DW000	%W000 - Low Word %W001 - High Word

## ⌘ 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 11000000 00000000 00000000 00000000 01000000 10000000 00000000 00000000 01000000 11000000 00000000 00000000 01000000 00100000 00000000 00000000	2  -2  4  6  2.5



## 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.



## Limitations



The Tsx47/Tsx67 driver works only in a Point to Point Mode, as the Protocol does not support Multi-Dropping.



## 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.

1. NULL Pointer for Login Data

3. NULL Pointer for Tag Table

**Explanation :** Internal Fatal Error.

**Action :** Contact Astra support.

2. NULL Pointer for Project Path

4. NULL Handle for Data Manager

5. Insufficient Memory for Request Manager

7. Insufficient Memory for Device Manager

**Explanation :** Internal Fatal Error.

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

6. Insufficient Memory for Transaction Manager

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.



## Errors



### 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.

### 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.



## Errors



- ⦿ 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. 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

**Explanation :** Internal Fatal Error.

**Action :** Contact Astra support.

### 31. Read Queue Full

### 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 (Applicable to S5).
- ⦿ Whether the Device model is the same as configured in the Node Configuration.



## Errors



- ⊙ 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.

### 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 150. If the queue is full this message will be prompted and the latest request will be ignored.

**Action** : Try configuring the Project such that at a time less than 150 write requests are raised. Also make sure that the Device gets enough time to serve these write requests.

### 35. 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.

### 36. Insufficient Memory for Register

### 37. Insufficient Memory for Tag Container2

### 38. Insufficient Memory for Register Container

**Explanation** : Internal Fatal Error.

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



## Errors



### 39. Tag Address Invalid

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

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

### 40. Tag Address Invalid2.

### 41. Driver Scan Halted

**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.

### 42. Driver Scan Halted2

### 43. 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.

### 44. 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 deinitialize itself.

**Action :** Unload Windows and restart again.

### 45. Cannot Run Without Initialisation

### 46. Cannot Run Without Initialization2

### 47. Cannot Write Without Initialisation

### 48. 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 deinitialize itself.

**Action :** Unload Windows and restart again.



## Errors



### 49. NULL Pointer for Queue

**Explanation** : Internal Fatal Error.

**Action** : Contact ASTRA support.

### 50. 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.

### 51. Cannot Pagelock Buffer

**Explanation** : Internal Fatal Error.

**Action** : Contact ASTRA support.

### 52. 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 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 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.



## Errors



### 53. 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.

### 54. Data Over Flow

**Explanation :** Unexpected data in large volume was received on the Communication port. It is also possible that the data entered by the user is out of range of the data type. This can happen for byte type of tags.

**Action :** Check –

- ⊙ If the cable connections have been disturbed.
- ⊙ If the Device has malfunctioned.
- ⊙ If the Communication hardware is proper and working.
- ⊙ Check the data entered by the user. In case of the byte type tags the data input range is between 0-255.

### 55. 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.



## Errors



56. Cannot Open File PLCTAG.DAT 2

57. Cannot Read File PLCTAG.DAT 2

58. 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.

59. Invalid BCD Format for a WORD

**Explanation**: The 16 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 plc.

60. Invalid BCD Format for a DWORD

**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 plc.

61. 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.

62. 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.

63. 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.



## Errors



### 64. 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.



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)



**RWTUV**