Communication Driver MPI

Driver for serial communication with Siemens devices using MPI interfaces

Index

1 Introduction

The MPI driver enables communication between Studio system and some of the Siemens devices using PRODAVE S7 software library, in accordance with the characteristics covered in this document.

This document contains 8 parts, as follows:

- **Introduction**: Provides an overview of the driver documentation.
- **General characteristics**: Provides information necessary to identify all the required components (hardware and software) necessary to implement the communication and global characteristics about the communication.
- **Installation**: Explains the procedures that must be followed to install the software and hardware required for the communication.
- **Driver configuration**: Provides the required information to configure the communication driver such as the different permutations for configuration and its default values.
- **Execution**: Explain the steps to test whether the driver was correctly installed and configured.
- **Troubleshooting**: Supplies a list of the most common error codes for this protocol and the procedures to fix them.
- **Application Sample**: Provides a sample application for testing the configuration the driver.
- **History of versions**: Provides a log of all the modifications done in driver.

✍ **Note:** This document presumes that the user has read the chapter *Driver Configuration* of the Studio's Technical reference manual.

2 General Characteristics

2.1 Device Characteristics

- **Manufacturer:** Siemens
- **Compatible Equipment:**
  - AS300, AS400, M7 and C7 from S7 series.
  - Any Siemens PLC from family S7 300/400 compatible with MPI protocol.
- **Siemens PLC programmer software:** Step7.

Tip: Refers to section 2.4 to see the Equipment used in the standard conformance tests for this driver.

2.2 Link Characteristics

- **Device communication port:** MPI Port
- **Physical protocol:** RS232/RS485
- **Logic protocol:** MPI Protocol
- **Device Runtime software:** Prodave S7
- **Specific PC Board:** Siemens MPI PC Boards or PC Adapters (USB and serial)
- **Adapters / Converters:** MPI

2.3 Driver Characteristics

- **Operating System:**
  - Windows XP

Tip: Please refer to section 2.4 to see the Operating System used in the conformance tests for this driver.

The driver is composed of the following files:

- **MPI.INI:** Internal file of the driver, it should not be modified by the user.
- **MPI.MSG:** Error messages for each error code. It should not be modified.
- **MPI.PDF:** Provides detailed documentation about the driver.
- **MPI.DLL:** Compiled driver.

✍ **Note:** All the files above must to be in the subdirectory /DRV of the Studio's installation directory.

✍ **Note:** It's necessary to install the PRODAVE software from Siemens to execute this driver. The Studio MPI driver uses the PRODAVE libraries to exchange data with the PLCs by MPI protocol.

- **Supported Registers:**

Register Type	Length	Write	Read	Bit	Integer	Float
M (Flags)	2 bytes	●	●	●	●	●
T (Timers)	2 bytes	–	●	●	●	–
Z (Counters)	2 bytes	–	●	●	●	–
E (Inputs)	1 byte	–	●	●	●	●
A (Outputs)	1 byte	●	●	●	●	●
DB (Data Blocks)	2 bytes	●	●	●	●	●

➲**Caution:** Bit writing is enabled in the Standard Driver Sheets only when using the **Write On Tag Change** command. In other words, it's not allowed execute bit writing by the command **Write Trigger** from the Studio Standard Driver Sheets.

2.4 Information about conformance testing

- **Equipment:** Siemens PLC model S7-300 CPU315-2DP
- **Configuration:**

- PLC Station: 4
- Baud Rate: 19,200
- Protocol: MPI
- Link: PC Adapter USB (6ES7 972-0CB20-0XA0)

- **Operating System (development):** Windows XP + Service Pack 3
- **Operating System (target):** Windows XP + Service Pack 3
- **Studio Version:** 7.0
- **Driver version:** 1.30

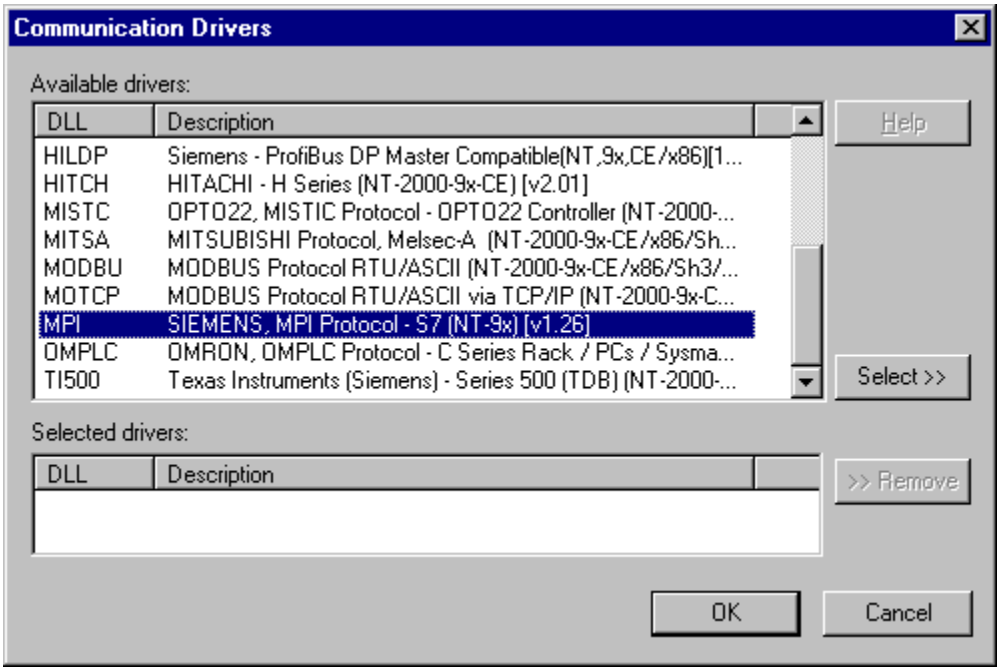
3 Installation

3.1 Installing the Driver

When you install the Studio v3.0 or higher, the communication drivers are already installed. You need now to select the driver at the applications where it will be used.

The steps to select the driver inside an application are:

1. Execute the Studio and select the proper application.
2. Select the menu *Insert + Driver...*
3. In the column **Available Drivers**, select the **MPI Driver** and push the button **Select >>** (the driver MPI must appear in the column **Selected Drivers**).
4. Press **OK**.



3.2 Other software requirements

It is necessary to install Prodave S7 software in the PC to enable the communication between the host and the Device. To download the custom program to the device, it is necessary to install one of the Siemens programmer software, for example, Step 7. Please see the Step 7 and Prodave S7 documentation about the procedure to install their software.

It's necessary to install the PRODAVE software from Siemens to execute this driver. The Studio MPI driver uses the PRODAVE libraries to exchange data with the PLCs by MPI protocol.

➲**Caution:** Special cautions must be taken when installing the physical hardware. Refer to the hardware manufacturer documentation for specific instructions in this area.

4 Driver Configuration

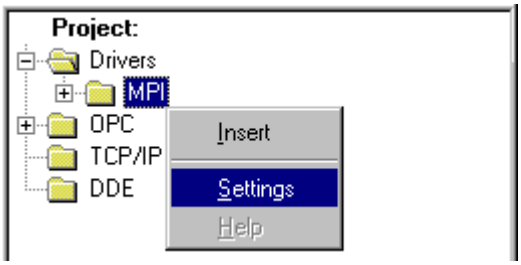
After the driver is installed and selected in the Studio (see section 3.1), you should proceed to the driver configuration.

The driver configuration is two parts:
- The Settings or Communication parameters, it is only one configuration to the whole driver;
- The communication tables or Driver Worksheets, where the communication tags are defined. There are two types of communication tables: **Standard Tables** and **MAIN DRIVER SHEET**.

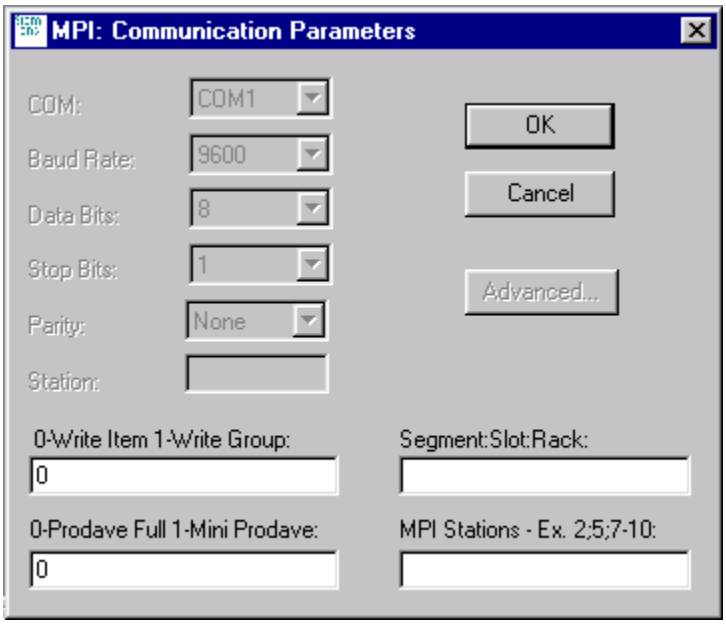
4.1 Settings - Communication Parameters

These parameters are valid for all driver worksheets configured in the system. To open the window for configuring the **Communication parameters**, follow these steps:

1. In the **Workspace** of the Studio environment, select the **Comm** table.
2. Expand the folder **Drivers** and select the subfolder **MPI**.
3. Right click on the **MPI** subfolder and select the option **Settings**.



When selecting the Settings, there is the following dialog to configure:



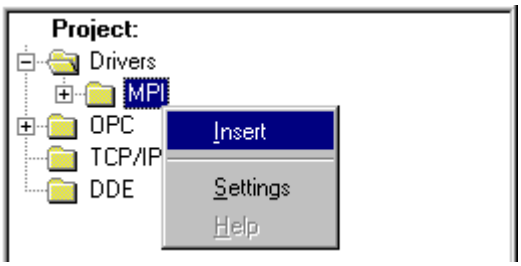
Parameter	Default Value	Valid values	Description
0-Write Item / 1-Write Item	0	1 or 2	Specify write item or write group when write trigger is triggered
Segment:Slot:Rack	0:2:0	See the Prodave S7 manual	Parameters used to initialize the MPI board.
0-Prodave Full 1 – Mini Prodave	0	0 or 1	Prodave S7 type
MPI Stations	0	-	Specify the stations that will exchange information with the driver's worksheets. You can type a single station(3) or a group(3-7) and both must be between semicolon Example: 1;3-7;20

✍ **Note:** The PRODAVE software MUST be configured with the SAME values defined in the **Communication Parameters** window of the MPI driver.

4.2 Driver Worksheet

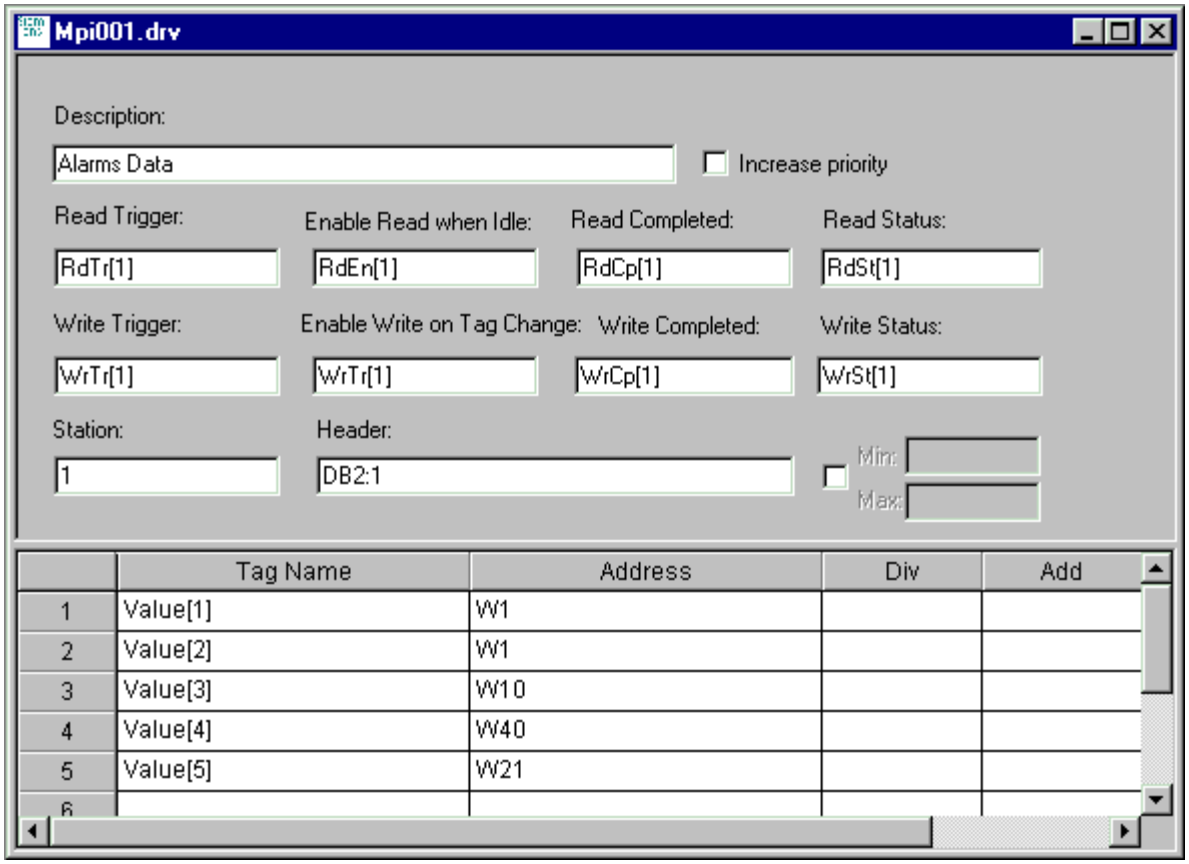
It is possible to configure many driver worksheets, each one will be composed of a Header and Body. To create a new driver worksheet, follow these steps:

1. In the **Workspace** of the Studio environment, select the table **Comm**.
2. Expand the folder **Drivers** and select the subfolder **MPI**.
3. Right click on the **MPI** subfolder and select the option **Insert**.



Tip: To optimize communication and ensure better performance for the system, it is important to tie the tags in different driver sheets according to the events that must trigger the communication of each group of tags and the periodicity for which each group of tags must be written or read. In addition, it is recommended to configure the addresses of communication in sequential blocks.

When creating a communication table, you have the following window:



All entries at the Driver Worksheet, exception by the **Station**, **Header** and **Address** are standard to all communication drivers. You should refer to Studio Communication Driver documentation about the configuration of the standard fields. This document describes the Station, Header and Address fields, which are specific to each communication driver.

4.3 Station and Header configuration

Parameter	Default Value	Valid values	Description
Station	-	1-255	PLC's Address.
Header	DB2	See next table	Defines the type of variable to be read or written from or to the device and the reference of the initial address.

The **Header** field defines the type of variables that will be read or written from or to the device. It complies with the syntax:

- To Flags, Timers, Counters, Inputs and Outputs :
  <Type>:<AddressReference>:<AddressNumberType> (e.g.: M:1:W)
- To Data-Blocks:
  <Type><TypeGroup>:<AddressReference>:<AddressNumberType> (e.g.: DB2:1:W)

- Type: Register type (M=Flags ; T=Timers ; Z=Counters ; E=Inputs ; A=Outputs ; DB=Data Blocks);
- TypeGroup: Group number of the register type configured (only for Data-Block types);
- AddressReference: Initial Address (reference) of the group configured. This number ALWAYS refers to the **Byte Address Number**. See next table;
- AddressNumberType*:* W means that the ***AddressOffset*** typed in the **Address** column is the amount of Words (offset) which will be added to the ***AddressReference***. B means that the ***AddressOffset*** typed in the **Address** column is the amount of Bytes (offset) which will be added to the ***AddressReference*** to compose the PLC address.

Header Address	Siemens Address		
Byte Address Number	**Byte Address Number**	**Word Address Number**	
Byte 0	Byte 0	W0	
Byte 1	Byte 1		W1
Byte 2	Byte 2	W2	
Byte 3	Byte 3		W3
Byte 4	Byte 4	W4	
Byte 5	Byte 5		W5
Byte 6	Byte 6	W6	
Byte 7	Byte 7		W7
Byte 8	Byte 8	W8	
Byte 9	Byte 9		W9
Byte 10	Byte 10	W10	
Byte 11	Byte 11		

Information regarding the parameter "Header"			
Type	**Sample of syntax**	**Valid range of initial Address**	**Comment**
Flags	M:1:W	Vary according to the equipment	Logical Flags.
Timers	T:2:W	Vary according to the equipment	Timer values.
Counters	Z:10:W	Vary according to the equipment	Counter values
Inputs	E:5:W	Vary according to the equipment	Physical input values
Outputs	A:8:W	Vary according to the equipment	Physical output values
Data Blocks	DB2:1:W	Vary according to the equipment	Data block Values, the number after "DB" (2) specify the data block number and the number after colon specify the word offset in the data block

4.4 Address Configuration

The body of the driver worksheet allows you to associate each tag to its respective address in the device. In the column **Tag Name,** you must type the tag from your application database. This tag will receive or send values from or to an address on the device. The address cells comply with the following syntax:

<Format><AddressOffset>.<Bit> (e.g.: W10.2)

- Format*:* Defines the treatment of the value read/written from/to the device. The current supported formats are:

B=Byte
W=Word
DW=Double Word
F=Float
T=Timer

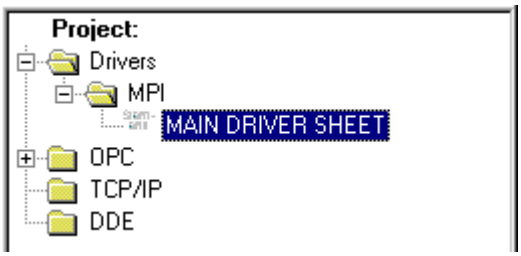
- AddressOffset*:* This parameter is added to the ***AddressReference*** (configured in the **Header** field) to compose the address of the group configure in the **Header** field. The ***AddressNumberType*** configured in the **Header** will define if the ***AddressOffset*** is a Byte offset or a Word offset;
- Bit*:* bit number (from 0 up to 15) from the word address. It's an optional parameter;

Sample of Addressing Configuration		
Address on the Device	**Header Field**	**Address Field**
M (Word 5 = Byte 5 / Byte 6)	M:0:B	W5
	M:5:B	W0
	M:1:W	W2
M (Byte 5)	M:0:B	B5
	M:5:B	B0
	M:1:W	B2
M (Byte 6)	M:0:B	B6
	M:6:B	B0
	M:0:W	B3
DB5 (Word 2 = Byte 2 / Byte 3)	DB5:0:B	W2
	DB5:2:B	W0
	DB5:0:W	W1
DB5 (Byte 2)	DB5:0:B	B2
	DB5:2:B	B0
	DB5:0:W	B1
DB5 (Byte 3)	DB5:0:B	B3
	DB5:3:B	B0
	DB5:1:W	B1
DB5 (Word 7 = Byte 7 / Byte 8)	DB5:0:B	W7
	DB5:7:B	W0
	DB5:1:W	W3
DB5 (Byte 7)	DB5:0:B	B7
	DB5:7:B	B0
	DB5:1:W	B3
DB5 (Byte 8)	DB5:0:B	B8
	DB5:8:B	B0
	DB5:0:W	B4

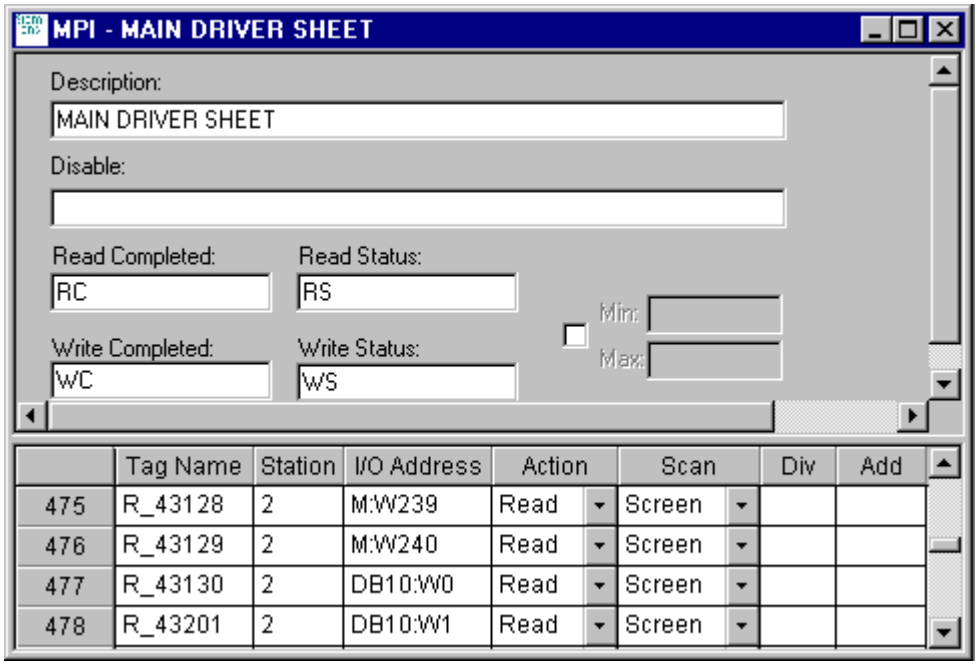
➲**Caution:** It's not allowed to configure in the same worksheet a range of addresses greater than the maximum block size (data buffer length) supported by the protocol: up to 1024 bytes in each Standard Driver Sheet.

4.5 Main Driver Sheet (MDS)

When the driver is inserted in the application, the MAIN DRIVER SHEET is automatically added to the driver folder.



The MAIN DRIVER SHEET provides a simple way to associate Studio tags to addresses in the PLC. Most of the MAIN DRIVER SHEET entries are standard for any driver. Refer to Studio Technical Reference Manual about the configuration of the standard fields. The fields which require specific syntax for this driver are described below:



	Tag Name	Station	I/O Address	Action	Scan	Div	Add
475	R_43128	2	M:W239	Read	Screen		
476	R_43129	2	M:W240	Read	Screen		
477	R_43130	2	DB10:W0	Read	Screen		
478	R_43201	2	DB10:W1	Read	Screen		

- **Station:** PLC's Address (ID number)

- **I/O Address:** Address of each register from the PLC. The syntax used in this field is described below:
  - To Flags, Timers, Counters, Inputs and Outputs :
    <Type>:<Format><Address>.<Bit> (e.g.: M:W3)

  - To Data-Blocks:
    <Type><TypeGroup>:<Format><Address>.<Bit> (e.g.: DB5:W10)

- *Type*: Register type
M=Flags
T=Timers
Z=Counters
E=Inputs
A=Outputs
DB=Data Blocks

- *TypeGroup*: Group number of the register type configured (only for Data-Block types);

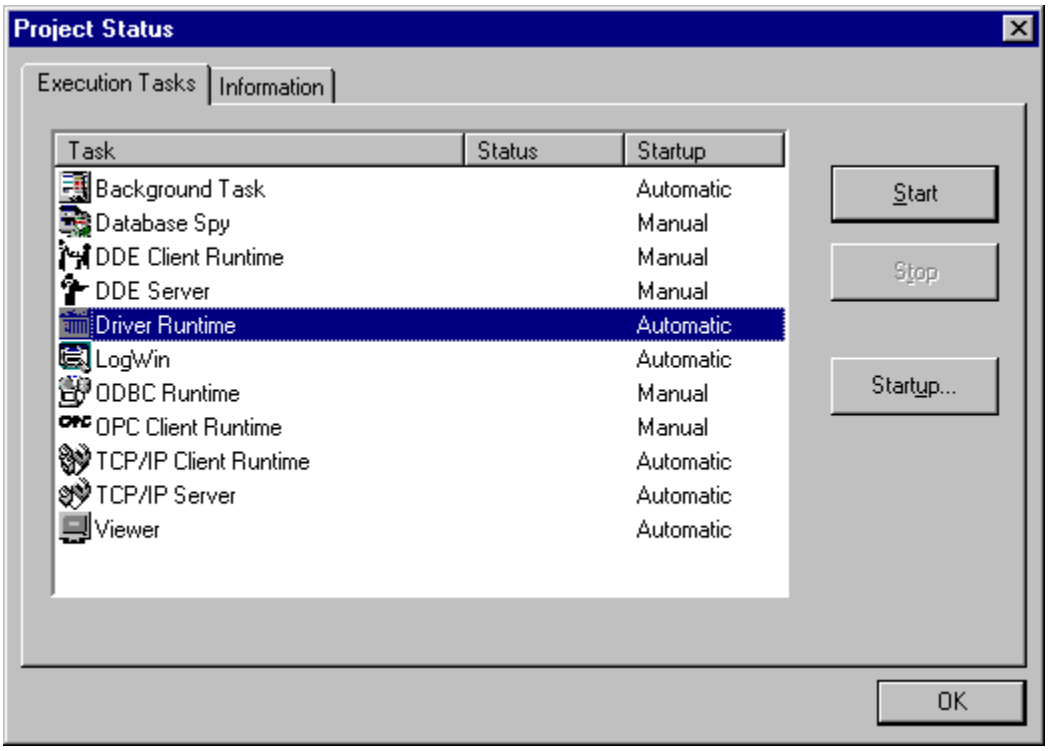
- *Format:*** Defines the treatment of the value read/written from/to the device. The current support formats are:

B=Byte
W=Word
DW=Double Word
F=Float
T=Timer

- *Address*: Address of the register. This number ALWAYS refers to the **Byte Address Number**;
- *Bit:*** bit number (from 0 up to 15) from the word address. It's an optional parameter;

5 Execution

When installing the driver, it is automatically selected to execute when you start-up the Runtime Environment. To verify if the driver is correctly enabled to start, use the menu option **Project** + **Status…,** and verify the task Driver Runtime



6 Troubleshooting

After each attempt to communicate using this driver, the tag configured in the field **Read Status** or **Write Status** will receive the error code regarding the kind of failure that occurred. The error messages are:

Error Code	Description	Possible causes	Procedure to solve
-15	Timeout waiting start a message.	- Disconnected cables - PLC turned off, or in Stop or error mode - Wrong Station number - Wrong RTS/CTS control settings.	- Check the cable wiring - Check the PLC state. It must be RUN - Check the station number. - Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.
1	Invalid Station	- Station is not configured on settings	- Check the settings for the station that is being used
2	Invalid Header	- An invalid header was set on header field or header part of main driver sheet address	- Check the headers and its syntax to comply with that described previously on this documentation
3	Invalid Address	- An invalid type for an address was set - Attempt to write on a bit of a Float	- Check the valid types and operations
4	Invalid Block Size	- The block size exceeds 1024 bytes	- Split the standard driver sheet that caused the error on more sheets
5	Write operation not supported	- Attempted to write on inputs, timers or counters headers. - Attempted to use write trigger to write on Bits	- Check headers for invalid write operations
10	PRODAVE DLL load error	- The PRODAVE DLL was not found on the system or its missing a dependency	- Check if PRODAVE is correctly installed
11	PRODAVE DLL functions error	- The PRODAVE DLL is not supplying the expected functions	- Check on the settings if the correct library is selected - Check if PRODAVE is correctly installed
12	Function not supported by PRODAVE Mini	- Attempt to read or write any header other than DB using the PRODAVE Mini	- Check the headers - Check if the settings are correct

➲**Caution:** Other error codes can be returned. The meaning of each one can be found in the PRODAVE S7 manual.

Tip: The communication status can be verified by the **output** Window of the Studio's environment or by the **Output Window**. To set a log of events for **Field Read Commands**, **Field Write Commands** and **Protocol Analyzer** click with the right button of the mouse on the output window and chose the option setting to select these log events.

When you are not able to establish the communication with the PLC, first of all establish the communication between the PLC Programming Tool and the PLC. Very frequently the communication is not possible due to a hardware or cable problem, or due an error or lack of configuration at the PLC. Only after the communication between the PLC Programming Software and the PLC is working fine, you can test again the supervisory driver.

When testing the communication with the Studio, you should first use the application sample described at item 7 (if it's available), instead of the new application that you are creating.

If is required to contact technical support, please have the following information available:

- Operating System (type and version): To find this information use the Tools/System Information option
- Project information: It is displayed using the option Project/Status from the Studio menu
- Driver version and communication log: Available from Studio Output when running the driver
- Device model and boards: please refer to hardware manufacture's documentation

7 Application Sample

There is currently no Sample Application for this driver

8 History of Versions

Revision	Version	By	Date	Description of changes
A	1.25	Lourenco	11-Dez-2000	▪ Version with the timer type
B	1.26	Roberto Vigiani Jr.	11-Jan-2001	▪ Included Byte Offset ▪ Included MAIN DRIVER SHEET
C	1.27	Eric Vigiani	30-Sep-2002	▪ Fixed bug reading Timer register
D	1.28	Leandro Coeli	05-May-2005	▪ Implemented Double Word ▪ Implemented Unsigned ▪ Fixed Bugs on DB reading
E	1.29	Diego Barros	16-Jan-2006	▪ Fixed bugs in write operations of Type F
F	1.30	André Körbes	27-May-2010	▪ Fixed bug with Main Driver Sheet and Double Word ▪ Updated documentation and troubleshooting section ▪ Added support for outputs writing
G	1.30	Andre Bastos	28-Jul-2011	▪ Documentation review only. No changes in the driver