AS511 Communication Driver

Driver for Serial Communication with
Siemens S5 Devices Using the AS511 Protocol

Contents

Introduction

The AS511 driver enables Serial communication between the Studio system and Siemens devices using the AS511 protocol, according to the specifications discussed in this document.

This document will help you to select, configure and execute the AS511 driver, and it is organized as follows:

- **Introduction**: This section, which provides an overview of the document.
- **General Information**: Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver**: Explains how to select the AS511 driver in the Studio system.
- **Configuring the Driver**: Explains how to configure the AS511 driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the AS511 driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Revision History**: Provides a log of all changes made to the driver and this documentation.

✍ **Notes:**

- This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Microsoft Windows NT/2000/XP environment. If you are not familiar with Windows, then we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter identifies all of the hardware and software components required to implement serial communication between the AS511 driver in Studio and a Siemens devices

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

Device Specifications

To establish communication, your target device must meet the following specifications:

- **Manufacturer:** Siemens
- **Compatible Equipment:** Siemens S5 Series CPU 943/944/945
- **Programmer Software:** Step 5

For a description of the device(s) used to test driver conformance, see “Conformance Testing”.

Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device Communication Port:** PG port
- **Physical Protocol:** RS232
- **Logic Protocol:** Siemens AS511
- **Device Runtime Software:** None
- **Specific PC Board:** None
- **Adapters/Converters:** None
- **Cable Wiring:** Cable must be compatible with Siemens S5 Series (e.g., *PC / RS232 Siemens Converter 6ES5 734-1BD20*).

Driver Characteristics

The AS511 driver package consists of the following files, which are automatically installed in the **`\DRV`** subdirectory of Studio:

- **`AS511.INI:`** Internal driver file. *You must not modify this file*.
- **`AS511.MSG:`** Internal driver file containing error messages for each error code. *You must not modify this file*.
- **`AS511.PDF:`** This document, which provides detailed information about the AS511 driver.
- **`AS511.DLL:`** Compiled driver.

✍ **Note:**
You must use Adobe Acrobat® Reader™ to view the **`AS511.PDF`** document. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe’s Web site.

You can use the AS511 driver on the following operating systems:

- Windows NT/2000/XP
- Windows CE

For a description of the operating systems used to test driver conformance, see “Conformance Testing” below.

The AS511 driver supports the following registers:

Register Type	Length	Write	Read	Bit	Byte	Word
E or I (*Inputs*)	1 Byte	–	●	●	●	●
A or Q (*Outputs*)	1 Byte	●	●	●	●	●
M or F (*Flags*)	2 Bytes	●	●	●	●	●
DB (*Data Blocks*)	2 Bytes	●	●	●	–	●
T (*Timer*)	2 Bytes	●	●	●	–	●
C or Z (*Counter*)	2 Bytes	●	●	●	–	●

Conformance Testing

The following hardware/software was used for conformance testing:

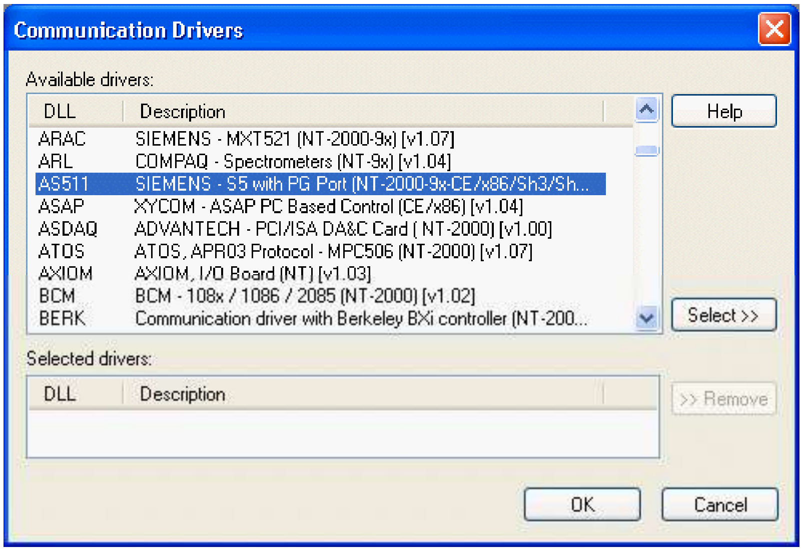
- **Equipment**: PLC S5 115U CPU944R
- **Driver Configuration**:
  - **Baud Rate**: 9600
  - **Protocol**: AS511
  - **Data Bits**: 8
  - **Stop Bits**: 1
  - **Parity**: Even
  - **COM Port**: COM1
- **Cable**: PC / RS232 Siemens Converter 6ES5 734-1BD20

Driver Version	Studio Version	Operating System (development)	Operating System (target)	Equipment
11.05	6.1 + SP1	Windows XP SP2	Windows XP SP2 Windows CE v5.00	PLC S5 115U CPU944R

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the **\DRV** subdirectory but they remain dormant until manually selected for specific applications. To select the AS511 driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **AS511** driver from the *Available Drivers* list, and then click the **Select** button.



DLL	Description
ARAC	SIEMENS - MXT521 (NT-2000-9x) [v1.07]
ARL	COMPAQ - Spectrometers (NT-9x) [v1.04]
AS511	SIEMENS - S5 with PG Port (NT-2000-9x-CE/x86/Sh3/Sh...
ASAP	XYCOM - ASAP PC Based Control (CE/x86) [v1.04]
ASDAQ	ADVANTECH - PCI/ISA DA&C Card (NT-2000) [v1.00]
ATOS	ATOS, APR03 Protocol - MPC506 (NT-2000) [v1.07]
AXIOM	AXIOM, I/O Board (NT) [v1.03]
BCM	BCM - 108x / 1086 / 2085 (NT-2000) [v1.02]
BERK	Communication driver with Berkeley BXi controller (NT-200...

DLL	Description

Communication Drivers Dialog

3. When the **AS511** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

> **Note:**
> It is not necessary to install any other software on your computer to enable communication between Studio and your target device. However, this communication can only be used by the Studio application; it cannot be used to download control logic to the device. To download control logic to a Siemens S5 Series device, you must also install the Siemens programming software (e.g., Step 5). For more information, please consult the documentation provided by the device manufacturer.

> **Attention:**
> For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

Configuring the Device

The *Siemens Step 5 Programmable Controller Manual* indicates the following serial communication settings:

- Baud Rate: 9600
- Data bits: 8
- Stop bits: 1
- Parity: Even
- Protocol: AS511
- Status: Run

Configuring the Driver

Once you have selected the AS511 driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver's communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

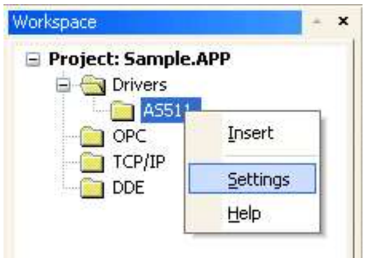
Configuring the Communication Settings

The communication settings are described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

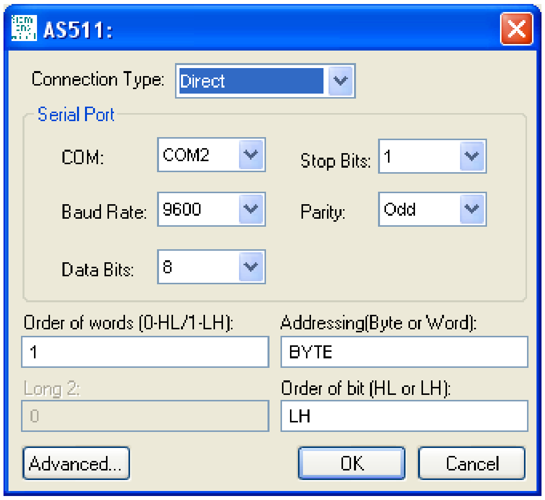
For the purposes of this document, only AS511 driver-specific settings and procedures will be discussed here. To configure the communication settings for the AS511 driver:

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The AS511 driver is listed here as a subfolder.

2. Right-click on the *AS511* subfolder and then select the **Settings** option from the pop-up menu. The *AS511: Communication Parameters* dialog is displayed:



Select Settings from the Pop-Up Menu



Connection Type: Direct

Serial Port

COM: COM2
Stop Bits: 1
Baud Rate: 9600
Parity: Odd
Data Bits: 8

Order of words (0-HL/1-LH): 1
Addressing(Byte or Word): BYTE
Long 2: 0
Order of bit (HL or LH): LH

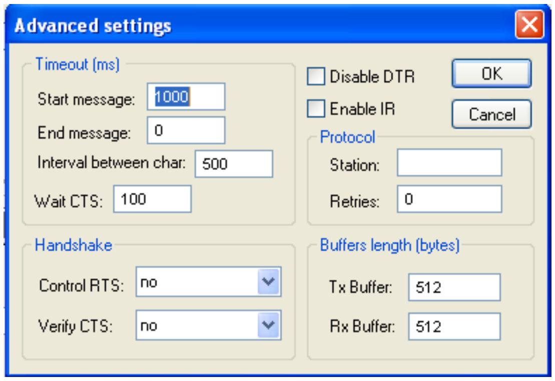
AS511: Communication Parameters Dialog

3. Verify the *Serial Port* settings, and change them if necessary.

4. Configure the additional driver-specific settings, as described in the following table:

Setting	Default Value	Valid Values	Description
Order of words	**1** (LH)	**0** (HL) or **1** (LH)	This setting applies only to Word writing and it determines the order in which Words (bytes) are written, from LOW to HIGH or from HIGH to LOW.
Addressing (Byte or Word)	**BYTE**	**BYTE** or **WORD**	Sets the type of register addressing, Byte or Word. For example, given a **Header** value of **F:0**, Byte addressing would address registers from **FY0** and Word addressing would address registers from **FW0**. NOTE: If you intend to use Data Blocks, then Word addressing is recommended.
Order of bit	**LH**	**LH** or **HL**	This setting applies only to Bit writing and it determines the order in which Bits are written, from LOW to HIGH or from HIGH to LOW. For example, given Word addressing and a Header value of **F:0**, LH would write **FY0 FY1** and HL would write **FY1 FY0**. NOTE: Bit *reading* is always determined by the target device, and on the 115U, it is always HL.

5. If you are using a Data Communication Equipment (DCE) converter (e.g., 232/485) between your PC and your target device, then you must also adjust the **Control RTS** (Request to Send) setting to account for the converter. In the *Communication Settings* dialog, click the **Advanced** button to open the *Advanced Settings* dialog:



Timeout (ms)
Start message: 1000
End message: 0
Interval between char: 500
Wait CTS: 100

□ Disable DTR
□ Enable IR

Protocol
Station:
Retries: 0

Handshake
Control RTS: no
Verify CTS: no

Buffers length (bytes)
Tx Buffer: 512
Rx Buffer: 512

Advanced Settings Dialog

When the dialog is displayed, configure the **Control RTS** setting using the following information:

Setting	Default	Values	Description
Control RTS	no	no	Do not set the RTS (Request to Send) handshake signal. IMPORTANT: If you are using Windows 95/98 or Windows CE with the correct RS232/RS485 adapter (i.e. without RTS control), then you must select this option.
		yes	Set the RTS (Request to Send) handshake signal before communication. IMPORTANT: If you are using Windows NT and the Cutler-Hammer RS232/RS485 adapter, then you must select this option.
		yes+echo	Set the RTS (Request to Send) handshake signal before communication, and echo the signal received from the target device.

➲ **Attention**:
If you incorrectly configure the **Control RTS** setting, then runtime communication will fail and the driver will generate a –15 error. See "Troubleshooting" for more information.

You do not need to change any other advanced settings at this time. You can consult the Studio *Technical Reference Manual* later for more information about configuring these settings.

6. Click **OK** to close the *Advanced Settings* dialog, and then click **OK** to close the *Communication Settings* dialog.

Configuring the Driver Worksheets

A selected driver includes one or more driver worksheets, which are used to associate database tags in Studio with operands on the target device. Each worksheet is triggered by specific application behavior, so that the tags / operands defined on that worksheet are scanned only when necessary – that is, only when the application is doing something that requires reading from or writing to those specific tags / operands. Doing this optimizes communication and improves system performance.

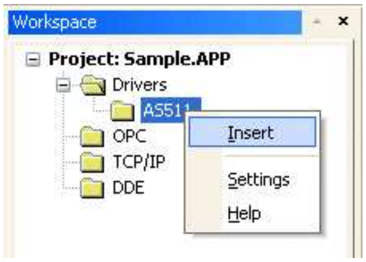
The configuration of these worksheets is described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

Note:
We recommend configuring operands on the device in sequential blocks in order to maximize performance.

To insert a new driver worksheet:

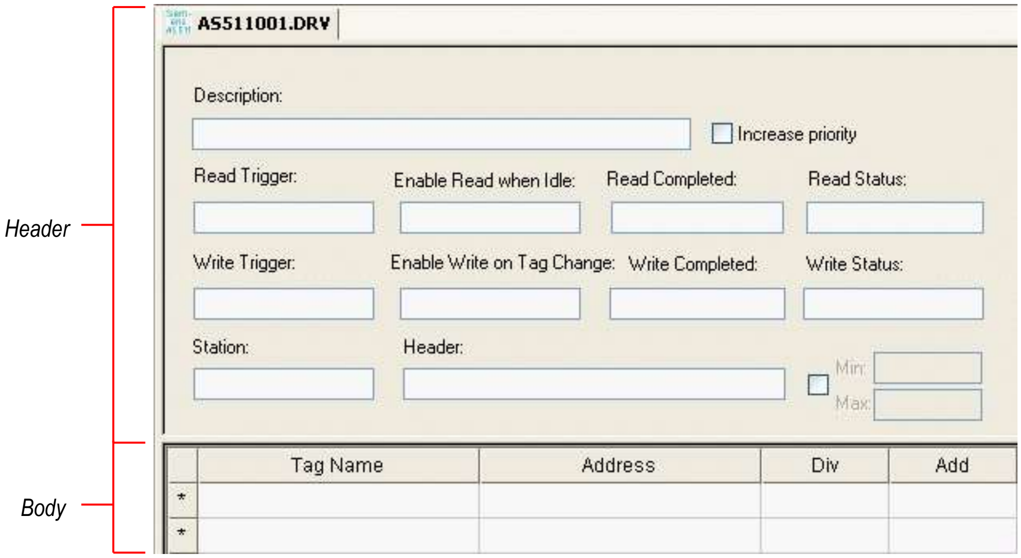
1. In the *Comm* tab, open the *Drivers* folder and locate the *AS511* subfolder.

2. Right-click on the *AS511* subfolder, and then select **Insert** from the pop-up menu:



Inserting a New Worksheet

A new AS511 driver worksheet is inserted into the *AS511* subfolder, and the worksheet is opened for configuration:



AS511 Driver Worksheet

> **Note:**
>
> Worksheets are numbered in order of creation, so the first worksheet is **`AS511001.drv`**.

Most of the fields on this worksheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the AS511 driver.

3. Configure the **Station** and **Header** fields as follows:

- **Station** field: Not used for Serial communication; you can configure the necessary settings in the *Communication Parameters* dialog described above.

- **Header** field: Specify the address of the first register of a block of registers on the target device. The addresses declared in the *Body* of the worksheet are simply offsets of this **Header** address. When Read/Write operations are executed for the entire worksheet (see **Read Trigger** and **Write Trigger** above), it scans the entire block of registers from the first address to the last.

  For Flags, Inputs and Outputs, use the following syntax:

  `<Register Type>[Data Format]:<Address Reference>`

  Examples — **`MW:1`**

  For Data Blocks, use the following syntax:

  `<Register Type><Block Number>:<Address Reference>`

  Example — **`DB2:1`**

Where:

– ***<Register Type>***: Siemens register type. Valid values are **E** or **I** (Input), **A** or **Q** (Output), **M** or **F** (Flag), **DB** (Data Block), **T** (Timer) and **C** or **Z** (Counter).

– ***[Data Format]*** (optional): Format of the data being read or written. Valid values are **B** (Byte) and **W** (Word). This parameter is *optional* and if it is omitted, then the value configured in **Addressing (Byte or Word)** is used by default. (See "Configuring the Communication Settings" above.)

Note:

The ***[Data Format]*** parameter does not apply to DB (Data Block), T (Timer), C or Z (Counter) registers; These registers are always formatted as Words.

– ***<Block Number>*** : Number of the data block.

– ***<Address Reference>***: The initial address (reference) of the block of registers configured on this worksheet. This value *always* refers to the Byte address number; see following table.

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		

For descriptions of valid values for the **Header** field, see the following table:

Type	Sample of syntax	Valid range of initial Address	Comment
Input	I:0, E:0, IB:0, IW:0, EB:0 or EW:0	Varies according to the equipment. (E.g. I0 to I127 for the 115U 944 PLC)	Physical Inputs. We can read its status from words or from bytes and we can, but should not, write on them.
Output	Q:0, A:0, QB:0, QW:0, AB:0 or AW:0	Varies according to the equipment. (E.g. Q0 to Q127 for the 115U 944 PLC)	Physical Outputs. We can read its status from words or from bytes and we can, but should not, write on them.
Flag	F:0, M:0, FY:0, FW:0, MY:0 or MW:0	Varies according to the equipment. (E.g. F0 to F255 for the 115U 944 PLC)	Internal Auxiliary Flag. Can be read or written by words, bytes or bits. Just take care about the **Order of bit writing** field at the Communication Parameters
Data Block	DB1:0	Varies according to the equipment. (E.g. DB0:0 to DB255:255 for the 115U 944 PLC)	We can read and write the whole data word from a Data Block and we can also read and write its block, always taking care with the "Order of bit writing" on the Communication Parameters window. In the Header, always type the DB number, followed by the initial data word number after the colon.
Timer	T:0	Varies according to the equipment.	We can read and write the whole data word, always taking care with the "Order of bit writing" on the Communication Parameters window.
Counter	C:0 or Z:0	Varies according to the equipment.	We can read and write the whole data word, always taking care with the "Order of bit writing" on the Communication Parameters window.

4. For each table row (i.e., each tag/register association), configure the **Address** field using the following syntax…

   <Address Offset>.[Bit]

   Example — **10**, **0.7**

Where:

– ***<Address Offset>*** : Value added to the ***<Address Reference>*** parameter (configured in the **Header** field above) to produce complete register address.

– ***[Bit]*** (optional): The bit number of the address. Valid values are **0** to **15** for Word-formatted addresses and Data Blocks, and **0** to **7** for Byte-formatted Inputs, Outputs and Flags.

For examples of how device registers are specified using **Header** and **Address**, see the following table:

Address on the PLC	Header Field	Address Field
IW0	I:0	0
IW10	I:0	10
IW10	I:10	0
I0.0	I:0	0.0
I0.7	IB:0	0.7
I1.0	IW:1	0.0
I1.5	I:1	0.5
QW0	Q:0	0
QW10	QW:0	10
QW10	Q:10	0
Q0.0	QB:0	0.0
Q0.7	QW:0	0.7
Q1.0	Q:1	0.0
Q1.5	Q:1	0.5
FW0	FW:0	0
FW10	F:0	10
FW10	F:10	0
F0.0	F:0	0.0
F0.7	FW:0	0.7
F1.0	FB:1	0.0
F1.5	F:1	0.5
DB100, DW 0	DB100:0	0
DB100, DW 100	DB100:0	100
DB100, DW 100	DB100:100	0
T10	T:0	10
T10	T:10	0
C5	C:5	0
C5	C:0	5

For more information about device registers and addressing, please consult the manufacturer's documentation.

➲ **Attention:**

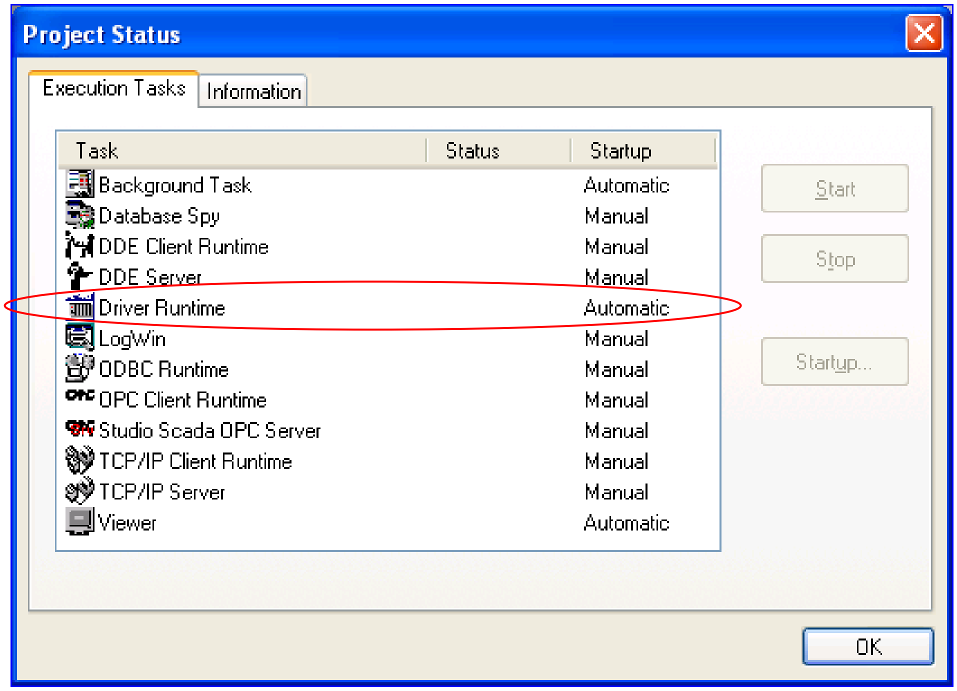
You must not configure a range of addresses greater than the maximum block size (data buffer length) supported by each device within the same worksheet. The maximum data buffer length for this driver is 255 bytes per driver worksheet.

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify that the the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Task	Status	Startup
Background Task		Automatic
Database Spy		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
LogWin		Manual
ODBC Runtime		Manual
OPC Client Runtime		Manual
Studio Scada OPC Server		Manual
TCP/IP Client Runtime		Manual
TCP/IP Server		Manual
Viewer		Automatic

Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
   - If the setting is correct, then proceed to step 3 below.
   - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the AS511 driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Main Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None.
1	Station number must always be 0	Wrong station number typed	Check on the driver configuration's worksheet that is getting this error if the configured station is different then. If so, correct it.
2	Only I, Q, F, DB1 – DB255 as allowed entries	An invalid Header has been typed or the tag that is inside this field has an invalid configuration.	Type a valid Header either on the header field or on the tag value. A lot of different valid headers are shown on the section IV.2
3	Only decimal values and bits allowed	An invalid Address has been typed.	Sometimes people type bit addressing with letters (0 to 9 and then A to F). This driver requires 0 to 15 or 0 to 7 as bit addressing. Take a look at the driver worksheet and correct it.
4	Block size exceeded	Block size is greater then 255	The highest allowed offset is 255. Reconfigure your Driver worksheet.
5	Protocol error	▪ Fail in the protocol ▪ Timeout waiting start a message ▪ Cable not connected ▪ PLC without a program ▪ Non-existent address in the PLC ▪ Incomplete answer sent by the PLC	There is something wrong with the communication. Check the communication parameters, the PLC settings, cable, etc… Sometimes the PLC does not send a complete answer when the message is too big.
6	Checksum error	Fail in the protocol	There is something wrong with the communication. Check the communication parameters and the PLC settings.
7	Non-existent module	Wrong header, station number, etc…	Check the Header and the values. It must be one of the valid.
8	Bit writing forbidden	It is not allowed to write bits.	Maybe we are trying to write a bit where it is not possible.
9	Write blocks only in continuous areas!	We are trying to write with the Write trigger and the worksheet has spaces between the addresses	Correct the worksheet without letting any space between the addresses. To write group all the addresses areas must be continuous.
10	Wrong Interface to the Transfer Mode	Problems with the serial ports	If you are using the transit mode with a number different then 0, so you are using another COM port. To get this error you must have declared a wrong COM port on the " Transit mode from COMx" Communication Parameters field.

AS511/S5-945 - Driver Configuration

1. Communication Parameters

On communication parameters, use:
Baud: 9600
Bits: 8
Stop: 1
Parity: Even

Modification in the AS511/S5-945 Drivers's Configuration (Version 9.21):
Addressing(Byte or Word): Addressing type of the protocol. Default: Byte
Order of bit writing(LH or HL): Order of bit writing (Low and High, or , High and Low). Default: LH

2. Read and Write groups

Station: not used
Header valid entries:

M	Flags
T	Timer
Z	Counter
E	Inputs
A	Outputs
DBxx	Data block xx

Address:
Offset
Offset.Bit //0..15 or 0..7 according to type

3. LogWin

Attention: when using this driver do not enable the LOG/SERIAL COMMUNICATION on LogWin, because it can generate communications errors with this drivers.

Modification in the AS511/S5-945 Drivers's Configuration (Version 8.00):

1. Offset Address

Now it is possible to define an offset address in the Header of the Driver's Configuration.
This new feature is very usefull in cases when it is necessary to do a shift in the address range in the PLC or if the user needs to realize an on-line driver configuration.
This modification was thought, in principle, for applications that need an on-line change of the offset-address.

In the former configuration, the user had to define in the HEADER field simply the data-range and the number of the data-block (DB), if the case.
Ex: Let's define the configuration-table for DB number **5**.
Header : DB5
Tag Name : var[0]

⇨ **Tip:**

You can monitor communication status by establishing an event log in Studio's *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Serial Communication,** right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the *LogWin* module (**Tools** → **LogWin**) to establish an event log on a remote unit that runs Windows CE. The log is saved on the unit in the `celog.txt` file, which can be downloaded later.

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software (e.g., Step 5). Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

If you must contact us for technical support, please have the following information available:

- **Operating System** (type and version): To find this information, select **Tools** → **System Information**.
- **Project Information**: To find this information, select **Project** → **Status.**
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Revision History

Revision	Version	Author	Date	Description of Changes
A	11.01	Roberto V. Junior	30-Jul-1999	▪ First driver version ▪ Driver available for Windows CE
B	11.02	Roberto V. Junior	19-Apr-2000	Implemented consistence tests when communicating with DBs.
C	11.03	José L. Teodoro	20-Jul-2001	Fixed problems with Windows CE.
D	11.04	José L. Teodoro	14-Nov-2001	Inserted the word order option.
E	11.05	Rafael R. Fernandes	22-Sep-2006	UDrvOpen Modified for SCPortOpen not to be called twice.
F	11.05	Michael D. Hayden	08-Dec-2006	Edited for language and usability.
G	11.05	Eric Vigiani	10-Mar-2008	Added information's about Timer and Counter.