Communication Driver ABKE

Driver for Serial Communication with Allen Bradley devices Using DF1 Protocol

Table of Contents

Introduction

The ABKE driver enables communication between the Studio system and some of the Allen Bradley devices using the DF1 protocol, if these components comply with the characteristics described in this document. You can use the ABKE driver to exchange data with PLCs through a DH+ network using the appropriate DF1→DH+ gateway.

This document was designed to help you install, configure, and execute the ABKE driver to enable serial communication with Allen Bradley devices. The information is organized, as follows:

- **Chapter 1 Introduction**: Provides an overview of the ABKE driver documentation.
- **Chapter 2 General Characteristics**: Describes all of the required hardware and software components you need to implement serial communication between the driver and the Allen Bradley devices. This chapter also discusses global characteristics about the communication.
- **Chapter 3 Installing the Driver**: Explains how to install the hardware and software components required for the ABKE driver.
- **Chapter 4 Configuring the Device**: Explains how to configure AB devices using the DF1 protocol.
- **Chapter 5 Configuring the Driver**: Explains how to configure the communication driver, including the different permutations for configuration and the driver's default values.
- **Chapter 6 Executing the Driver**: Explains how to execute the driver to verify that you installed and configured the driver correctly.
- **Chapter 7 Troubleshooting**: Lists the most common error codes for this protocol and explains how to fix the errors.
- **Chapter 8 Using the Application Sample**: Provides a sample application, which you can use to test the driver configuration.
- **Chapter 9 Revision History**: Provides a log of all modifications made to the driver.
- **Appendix: Setting Up the PLC Communication with the Manufacturer Programming Software**: Explains how to use Rockwell RSLogix software to ensure that the PLC, connections, and cables are functioning properly before using Studio.

✎ **Note:** This document assumes that you have read the "Driver Configuration*"* chapter of the *Studio Technical Reference Manual*.

This document also assumes that you are familiar with the Windows NT/2000 environment. If you are unfamiliar with Windows NT/2000, we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Characteristics

This chapter discusses the characteristics of the hardware and software components used to implement communication between Allen Bradley devices and the ABKE driver. In addition, this chapter provides information about the equipment used for conformance testing.

The chapter is organized as follows:

- Device Characteristics
- Link Characteristics
- Driver Characteristics
- Conformance Testing

Device Characteristics

To establish serial communication with the ABKE driver, you must use devices with the following specifications:

- **Manufacturer**: Allen Bradley

- **Compatible Equipment**:
  - PLC2 series
  - PLC5 series
  - SLC500
  - MicroLogix 1100/1200/1500 (compatible with SLC500 series)

- **Rockwell PLC Programmer Software**:
  - RSLogix
  - ASP
  - RS6200

> **Tip:** Refer to the "Conformance Testing" section to review the equipment used in the standard conformance tests for this driver.

Link Characteristics

To establish serial communication between Allen Bradley devices and the ABKE driver, you must follow these specifications:

- **Device Communication Port**: Serial DF1 port
- **Physical Protocol**: Serial RS232
- **Logic Protocol**: DF1
- **Device Runtime Software**: None
- **Specific PC Board**: None
- **Cable**: Allen Bradley 1756CP3A/A01 S760 KSM

Driver Characteristics

The ABKE driver is composed of the following files:

- **ABKE.INI:** Internal file of the driver. *You must not modify this file*.
- **ABKE.MSG:** Error messages for each error code. *You must not modify this file*.
- **ABKE.PDF:** Document providing detailed information about the ABKE driver.
- **ABKE.DLL:** Compiled driver.

Notes: All of the preceding files have been installed in the `/DRV` subdirectory of the Studio installation directory.

You must use Adobe Acrobat Reader (provided with the Studio installation package) to view the ABKE.PDF document.

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

- Windows NT/2000/XP
- Windows CE

Tip: Refer to the "Conformance Testing" section to review the operating systems used in the standard conformance tests for this driver.

This driver supports the following Register types:

Register Type	Length	Default Value	Write	Read	Bit	Word	Float	String	BCD
O (*Output*)	2 Bytes	W	•	•	•	•	–	•	•
I (*Input*)	2 Bytes	W	•	•	•	•	–	•	•
S (*Status*)	2 Bytes	W	•	•	•	•	–	•	•
B (*Binary*)	2 Bytes	W	•	•	•	•	–	•	•
T (*Timer*)	6 Bytes	W	•	•	–	•	–	–	–
C (*Counter*)	6 Bytes	W	•	•	–	•	–	–	–
R (*Control*)	6 Bytes	W	•	•	–	•	–	–	–
F *(Float)*	4 Bytes	F	•	•	–	–	•	–	–
N *(Integer File)*	2 Bytes	W	•	•	•	•	–	•	•
ST *(String File)*	N Bytes	S	•	•	–	–	–	•	–
A *(ASCII File)*	N Bytes	S	•	•	•	•	–	•	•

Attention: The Format B option (BCD) applies to the first 12 bits only. You can view the last 4 bits in the "`Quality`" field. For example: When reading the `TAG_N7[0]` tag from the `N7:0` address, the last 4 bits of this address are written to the `TAG_N7[0]->Quality` field.

Attention: Float format uses 4 bytes (2 Words). When using the Float format with a register type that has the default size in 2 bytes. Then, it you use two consecutives addresses. This happens with Output, Input, Status, Binary, Integer and ASCII.

Conformance Testing

The following equipment was used for the conformance testing:

- **Equipment**: PLC5/40 and SLC500/03
- **Configuration:**
  - Baud Rate: 19200
  - Protocol: DF1
  - Data Bits: 8
  - Stop Bits: 1
  - COM port: COM1

- **Cable**: According to specifications described in the “Link Characteristics” section
- **Operating System (development)**: Windows XP + Service Pack 2
- **Operating System (target)**:
  - Windows XP + Service Pack 2
  - Windows CE v4.2

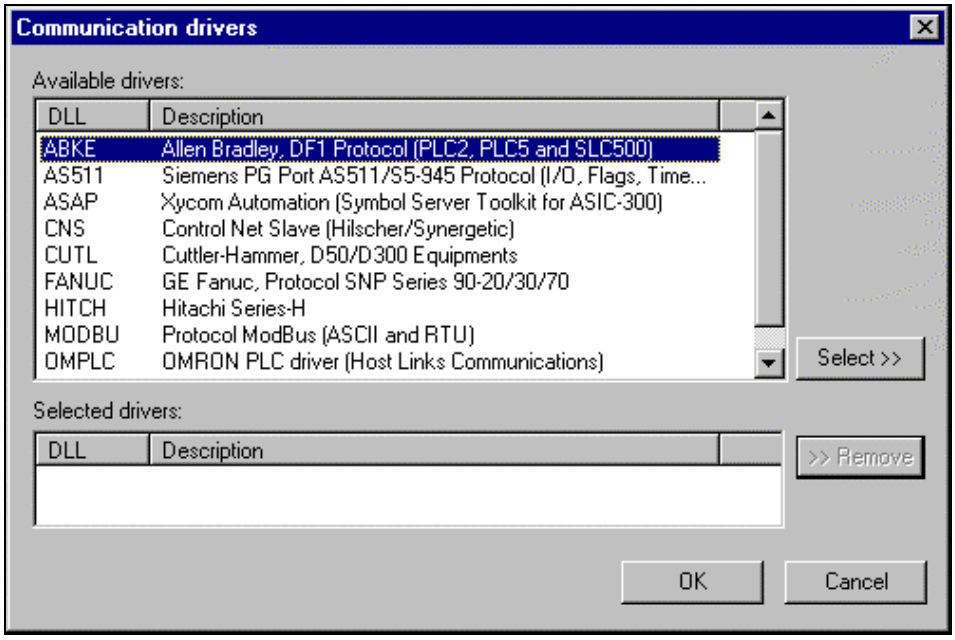
- **Studio Version**: 8.0 + SP1
- **Driver Version**: 10.6

Installing the Driver

When you install Studio v3.0 and higher, all of the communication drivers are installed automatically. You must select the driver that is appropriate for the applications you are using.

Perform the following steps to select the driver from within an application:

1. Execute Studio using one of the following methods:
   - Double-click the **Studio** shortcut icon from the desktop.
   - Click on the **Start** menu, select **Programs**, and when the **Studio Tools** submenu displays, select **Studio**.
2. When the *Studio* window opens, open the appropriate application from the Workspace pane.
3. From the main menubar, select **Driver…** from the **Insert** menu to open the *Communication drivers* dialog box as shown in the following figure.



4. Select the **ABKE driver** from the **Available Drivers** list and click the **Select>>** button.
5. When the ABKE driver displays in the **Selected Drivers** list, click the **OK** button to close the dialog.

You are not required to install any other software on your computer to enable communication between Studio and the Allen Bradley device. However, you must install one of the Rockwell programmer software packages (such as RSLogix) to download the custom program to this device. Consult the Rockwell RSLogix documentation for a description of the procedure you must use to install their software.

➲**Attention:** You must use special precautions when installing the physical hardware. Refer to the hardware manufacturer's documentation for detailed instructions.

Configuring the Device

You can use one of several methods for configuring the CPU serial channel. However, the default configuration for a PLC is as follows:

- **Baud Rate**: 19200
- **Data Bits**: 8
- **Stop Bits**: 1
- **Parity**: None
- **Error Check**: BCC

Configuring the Driver

After you install the ABKE driver and open Studio (as described in Chapter 3), you can configure the driver.

You configure the driver in two stages:

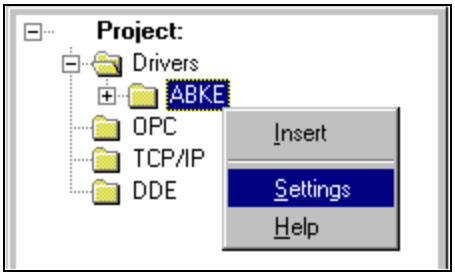
- You set the communication parameters (only one configuration for the whole driver).
- You define the communication tags by completing the Driver Worksheets. There are two types of Driver Worksheets:
    - Standard Driver Worksheet
    - MAIN DRIVER SHEET (MDS)

The following sections provide instructions for setting the parameters and completing the worksheets.

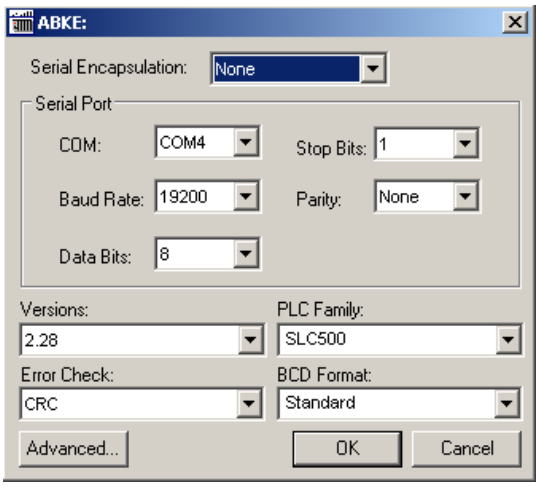
Setting the Communication Parameters

When you set the communication parameters, they are valid for all Driver Worksheets configured in the system. Use the following steps to configure the communication parameters for the driver:

1. From the *Studio* application window, click the **Comm** tab located below the Workspace pane.
2. From the Workspace pane, expand the **Drivers** folder.
3. Right-click on the **ABKE** subfolder and when the pop-up menu displays, (as shown in the following figure) select the **Settings** option.



The *Communication Parameters* dialog displays as follows:



4. You must configure the following parameters:

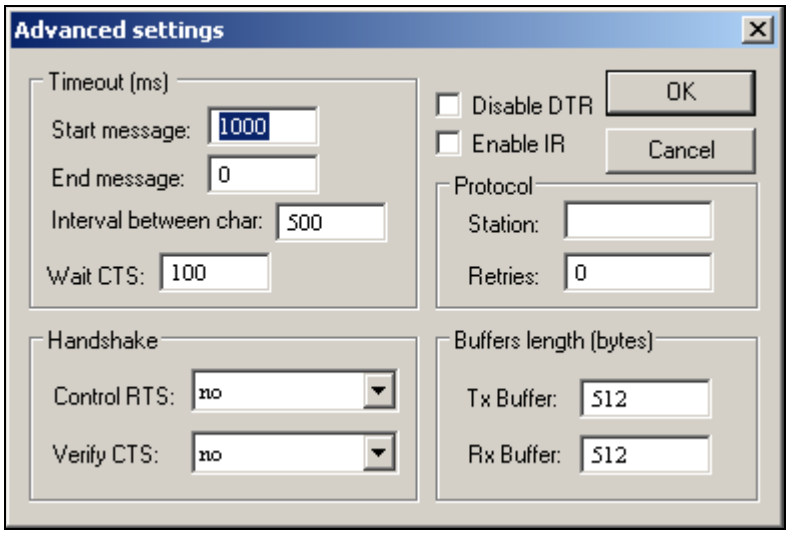
Parameter	Default Value	Valid Values	Description
Versions	**v2.28**	**v2.27** or **v2.28**	The option **v2.28** enables the ASCII functionalities working correctly. The **v2.27** was kept only for compatibility reasons.
PLC Family	**PLC2**	**PLC2**, **PLC5**, **PLC5 with I/O Octal** or **SLC500**	Select the PLC Family that connects to the driver. **PLC2:** Driver uses "unprotected Read/Write" command (in the DF1 protocol) to communicate with the PLC2 family or compatible. **PLC5:** Driver uses "typed Read/Write" command (in the DF1 protocol) to communicate with the PLC5 family or compatible. The address for all data types is decimal. **PLC5 with I/O Octal:** Driver uses "typed Read/Write" command (in the DF1 protocol) to communicate with the PLC5 family or compatible. The address for I and O data types is octal. The address for all remaining data types is decimal. **SLC500:** Driver uses "protected typed logical Read/Write" command (in the DF1 protocol) to communicate with the SLC500 family or compatible.
Error Check	**BCC**	**BCC** or **CRC**	Defines the error check method that the driver will use.
BCD Format	**Legacy**	**Legacy or Standard**	**Legacy:** Support for old version, three digits will be stored in the tag value for BDC and the fourth digit will be stored in the quality field. This is done to keep compatibility with old 16-bit versions of the product. **Standard:** Stored all the BDC digits in the tag value.

PLC Family field - Legacy Compatible	
New	**Old**
PLC2	`2`
PLC5	`5 or 5:0 or 5x`
PLC5 with I/O Octal	`5:1`
SLC500	`500`

✍ **Note 1:** To communicate with PLC 5 family, we strongly recommend selecting the option **PLC5 with I/O Octal** instead of **PLC5** in the Family field. The option **with I/O Octal** allows access to the I/Os in Octal, matching the PLC addressing mode, as well as direct access to the **ASCII (A)** file type by its word address.

✍ **Note 2:** You *must* configure the Allen Bradley device with the *same* values defined for the ABKE driver in the *Communication Parameters* dialog.

5. Click on the **Advanced…** button in the *Communication Parameters* dialog. The *Advanced settings* dialog will display.



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.

✍ **Note**: Although you can configure other serial communication parameters from this dialog, you should not change any of the default parameters at this time except **Control RTS**. The parameters on the *Advanced settings* dialog are explained in detail in the *Studio Technical Reference Manual*.

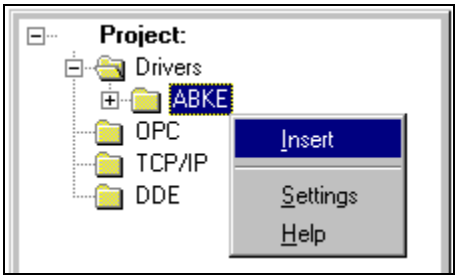
Tip: Usually, you must change these parameters if you are using a DCE (Data Communication Equipment) converter (232/485, for example), modem, and so forth between the PC, driver, and the host. You must know the characteristics of the DCE before adjusting these parameters.

Configuring the Standard Driver Worksheet

This section explains how to configure a Standard Driver Worksheet to define communication tags. You can configure multiple Driver Worksheets, each of which is divided into a Header and a Body.

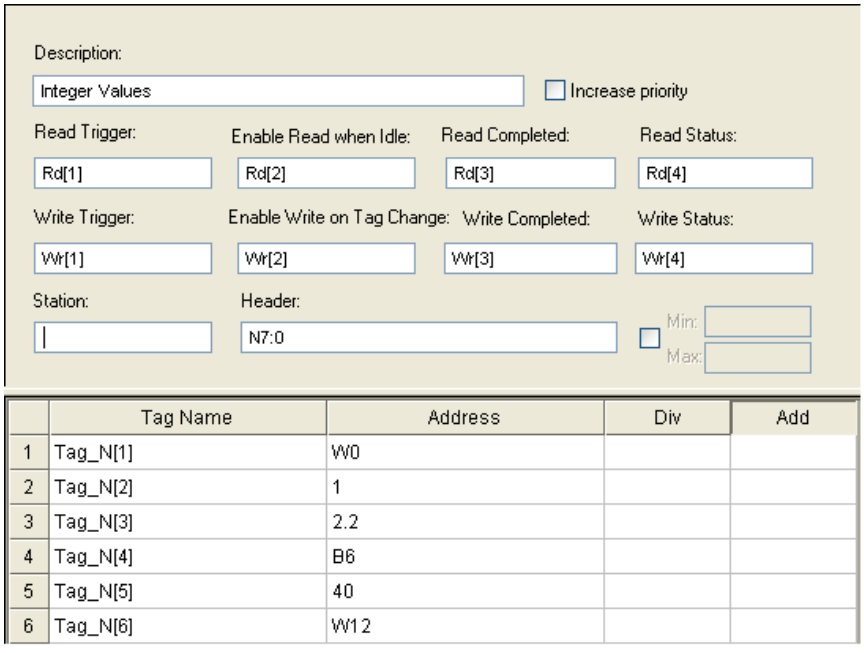
Use the following steps to create a new Standard Driver Worksheet:

1. From the *Studio* application window, select the **Comm** tab, located below the Workspace pane.
2. In the Workspace pane, expand the **Drivers** folder and right-click the **ABKE** subfolder.
3. When the pop-up menu displays (as shown in the following figure), select the **Insert** option.



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

The *ABKE001.drv* worksheet displays (similar to the following figure).



	Tag Name	Address	Div	Add
1	Tag_N[1]	W0		
2	Tag_N[2]	1		
3	Tag_N[3]	2.2		
4	Tag_N[4]	B6		
5	Tag_N[5]	40		
6	Tag_N[6]	W12		

All fields on the Standard Driver Worksheet are standard for all communication drivers; except for the **Station**, **Header,** and **Address** fields. This document explains only the Station, Header, and Address fields because they are specific to each communication driver. For detailed information about the configuring of the standard fields refer to the *Studio Technical Reference Manual*.

Proceed to the next section for an explanation about configuring the **Station** and **Header** fields.

Configuring the Station and Header Fields

This section explains the procedure for configuring the Station and Header fields. The following table describes the default and valid values for these two fields:

Parameter	Default Value	Valid Values	Description
Station	-	1 to 31	PLC's Address
Header	N7:0	See next table	Defines the type of variable to be read or written from, or to, the device and references the initial address

When specifying the Header parameter, you must comply with the following syntax:

- For **Input and Output**: `<Type>:<SlotNumber>.0` (for example: `O:1.0`)
- For **Status**, **Binary**, **Integer, Float, Timer**, **Counter**, **Control, String and ASCII**:

  `<Type><TypeGroup>:<AddressReference>` (for example: `N7:0`)

  Where:

  - ***Type*** (Register Type): Type one of the following O=Output, I=Input, S=Status, B=Binary, N=Integer, T=Timer, C=Counter, R=Control, F=Float, A=ASCII, ST=String.
  - ***SlotNumber***: Type the I/O Card Slot Number.
  - ***TypeGroup***: Type the Group Number of the register type you configured.
  - ***AddressReference***: Type the Initial Address (reference) of the group you configured.

After you edit the **Header** field, the system checks that the syntax is valid. If the syntax is incorrect, the system automatically inserts the default value (N7:0) into the Header field.

If you type a Tag string between curly brackets `{Tag}` into this field, you must ensure that the Tag value and syntax are both correct or an `Invalid Header` error will result. The following table describes the proper syntax for both the field type and the Tag value.

Information about the Header Parameter			
Field Type	**Syntax Sample**	**Valid Range for Initial Address**	**Comments**
Output	O:0.0	Varies according to the equipment	Physical outputs: Where "O" means output. The first digit after the colon defines the word number if there is more than one digit in the same slot and the first digit following the dot is the output address.
Input	I:0.0	Varies according to the equipment	Physical inputs: Where "I" means input. The first digit after the colon defines the word's number if there's more than one digit in the same slot and the digit following the dot is the output address.
Status	S:0	Varies according to the equipment	Reads the status words.
Binary	B3:0	0 to 255	Reads the binary operator.
Integer	N7:0	0 to 255	Reads and Writes the Integer addresses.
Timer	T4:0	0 to 255	Reads and Writes the Timer addresses.
Counter	C5:0	0 to 255	Reads and Writes the Counter addresses.
Control	R6:0	0 to 255	Reads and Writes the Control addresses.
Float	F8:0	0 to 255	Reads and Writes the Float addresses.
ASCII	A14:0	0 to 255	Reads and Writes the ASCII addresses. **SLC500:** The "Address Reference" is defined in words. **PLC5:** The "Address Reference" is defined in bytes.
String	ST15:0	0 to 255	Reads and Writes the String addresses.

After specifying the **Station** and **Header** field parameters, proceed to the next section for instructions about configuring the **Address** field.

Configuring the Address Field

The body of the Driver Worksheet allows you to associate each tag to its respective address in the device. In the **Tag Name** column**,** you must type the tag from your application database. This tag receives or sends values from, or to, an address on the device. The address field must comply with the following syntax:

- For **Input** and **Outputs**: `[Format]<OctetNumber>/[Bit]` (for example: `W0/3`)
- For **Status**, **Binary**, and **Integer**: `[Format]<AddressOffset>/[Bit]` (for example: `W10/12`)
- For **Timer**, **Counter**, and **Control**: `[Format]<AddressOffset>.<Element>` (for example: `W2.PRE`)
- For **Float**: `[Format]<AddressOffset>` (for example: F6)
- For **String**: `[Format]<AddressOffset>.<Number of Bytes>` (for example: S1.2)

- For **ASCII**:
  - `[Format]<AddressOffset>.[Number of Bytes]` (for example: `S1.2`)
  - `[Format]<AddressOffset>` (for example: `W0`)

  Where:

  - ***Format***: Type *(optional parameter)*

    `-W` to treat the values as words
    `-B` to treat the values as BCDs
    `-F` to treat the values as Floats (2 words)

`-S` to treat the values as Strings

- ***OctetNumber***: Type the Octet Number of the I/O card you configured in the **Header** field.
- ***AddressOffset****:* Add this parameter to the AddressReference (configured in the **Header** field) to compose the address of the Group you configured in the **Header** field.
- ***Number of Bytes****:* Specify the Maximum Size of an ASCII/STRING data type.
- ***Address****:* Type the address of the Group you configured in the **Header** field.
- ***Bit***: Type the Bit Number (from 0 to 15) from the word address. (*optional parameter*)
- ***Element****:* Specify the Element Type for Timers, Counters, and Controls according to the following table:

Register	Elements																	
	DN	PRE	ACC	EN	TT	UA	UN	OV	CD	CU	FD	IN	UL	ER	EM	EU	LEN	POS
Timer	●	●	●	●	●	–	–	–	–	–	–	–	–	–	–	–	–	–
Counter	●	●	●	–	–	●	●	●	●	●	–	–	–	–	–	–	–	–
Control	●	–	–	●	–	–	–	–	–	–	●	●	●	●	●	●	●	●

✎ **Note:** The String Format needs to have the size of the String. The size 2 is the size default.

➲**Attention:** You can use a group of tags for writing, but remember that This command always will write zero in the Addresses between the maximum and minimum that are not configured in the associated worksheet.

➲**Attention:** The Format B option (BCD) applies to the first 12 bits only. You can view the last 4 bits in the "`Quality`" field. For example: When reading the `TAG_N7[0]` tag from the `N7:0` address, the last 4 bits of this address are written to the `TAG_N7[0]->Quality` field.

➲**Attention:** For worksheets that handles ASCII data type (**A** type in the Header and **S** type in the Address fields)
In PLC5 (assuming **5x** was chosen in Family parameter) you'll be using "Typed Read/Write" commands. With this command you may read starting from an odd address, but you cannot write to an odd address.
The final address is given by the sum of the initial address in the Header field and the least address found in the Address column of the worksheet.
When writing ASCII data type, be sure to write to even addresses.

Sample of Address Configuration		
Address on the Device	**Header Field**	**Address Field**
I:0/7	I:0.0	W0/7 or 0/7
I:0/10	I:0.0	W0/10 or 0/10
I:0/17	I:0.0	W0/17 or 0/17
I:0/25	I:0.1	W0/9 or 0/9
I:3/4	I:3.0	W0/4 or 0/4
I:0/4	I:0.0	W0/4 or 0/4
O:0/7	O:0.0	W0/7 or 0/7
O:0/10	O:0.0	W0/10 or 0/10
O:0/17	O:0.0	W0/17 or 0/17
O:0/25	O:0.1	W0/9 or 0/9
O:3/4	O:3.0	W0/4 or 0/4
O:0/4	O:0.0	W0/4 or 0/4
S:0/5	S:0	W0/5 or 0/5
S:10/7	S:0	W10/7 or 10/7
S:10/7	S:10	W0/7 or 0/7
B3:0/5	B3:0	W0/5 or 0/5
B3:10/7	B3:0	W10/7 or 10/7

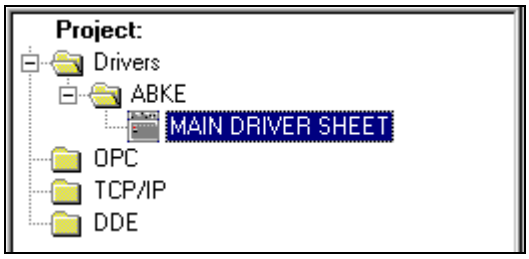
B3:10/7	B3:10	W0/7 or 0/7
N7:0	N7:0	W0 or 0
N7:0/10	N7:0	W0/10 or 0/10
N7:50	N7:20	W30 or 30
T4:0/ACC	T4:0	W0/ACC or 0/ACC
T4:0/PRE	T4:0	W0/PRE or 0/PRE
T15:0/EN	T15:0	W0/EN or 0/EN
T15:0/ACC	T15:0	W0/ACC or 0/ACC
T15:1/ACC	T15:0	W1/ACC or 1/ACC
C5:0/ACC	C5:0	W0/ACC or 0/ACC
C5:1/PRE	C5:0	W1/PRE or 1/PRE
C20:15/UA	C20:10	W5/UA or 5/UA
R6:0/LEN	R6:0	W0/LEN or 0/LEN
R6:0/POS	R6:0	W0/POS or 0/POS
R6:1/POS	R6:0	W1/POS or 1/POS
F8:0	F8:0	F0 or 0
F8:5	F8:5	F0 or 0
F8:5	F8:0	F5 or 5
A14:0 (default size: 2 bytes)	A14:0	S0.2 or 0.2
A14:1 (default size: 2 bytes)	A14:1	S0.2 or 0.2
A14:1 (default size: 2 bytes)	A14:0	S1.2 or 1.2
A14:0 to A14:2	A14:0	S0.6 or 0.6
ST15:0 (String: maximum 20 bytes)	ST15:0	S0.20 or 0.20
ST15:1 (String: maximum 50 bytes)	ST15:0	S1.50 or 1.50
ST15:2 (String: maximum 10 bytes)	ST15:1	S1.10 or 1.10

➲**Attention:** You are not permitted to configure a range of addresses greater than the maximum block size (data buffer length) supported by each PLC (as follows) in the same worksheet:

- For PLC 2 and PLC 5 devices, the maximum data buffer length is 244 bytes.
- For the SLC device family in the Read commands, the maximum data buffer length is 236 bytes (SLC.03 and SLC.04) and 82 bytes (SLC5.01 and SLC5.02).
- For the SLC device family in the Write commands, the maximum data buffer length is 234 bytes (SLC.03 and SLC.04) and 82 bytes (SLC5.01 and SLC5.02).

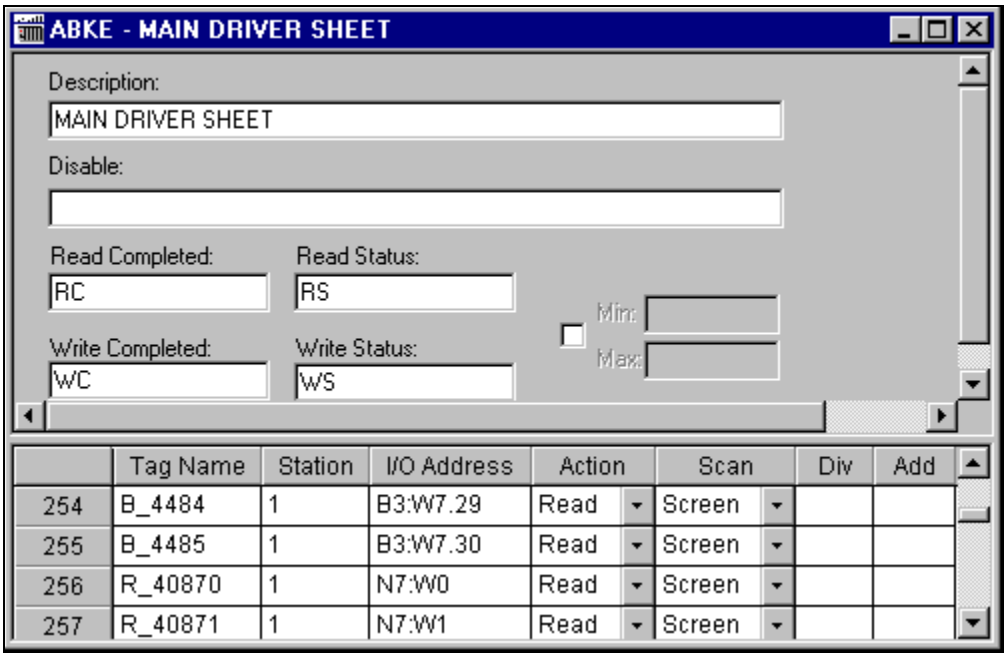
Configuring the MAIN DRIVER SHEET (MDS)

When you add the ABKE driver to your application, the program automatically adds the MAIN DRIVER SHEET (*MDS*) to the **ABKE** driver folder (refer to the following figure).



The MDS provides a simple way for you to associate Studio tags to addresses in the PLC. Most MDS entries are standard for any driver. For detailed information about configuring these standard entries, refer to the *Studio Technical Reference Manual*.

1. Double-click on the **MAIN DRIVER SHEET** icon to open the following worksheet:



2. Complete the fields on this dialog, being sure to comply with the following syntax:

   - **Station:** PLC Address (ID number)
   - **I/O Address:** Address of each register from the PLC.

     - For Input and Outputs: `<Type>:<SlotNumber>.[Format]<OctetNumber>.[Bit]` (for example: `O:1.W2.4`)
     - For Status, Binary and Integer: `<Type><TypeGroup>:[Format]<Address>.[Bit]` (for example: `N7:W150.2`)

- For Timer, Counter and Control: `<Type><TypeGroup>:[Format]<Address>.<Element>` (for example: `T4:W0.DN`)
- For ASCII and String: `<Type><TypeGroup>:[Format]<Address>.[Number of Bytes]` (for example: `ST15:S0.50`)
- For Float: `<Type><TypeGroup>:[Format]<Address>`

Where:

- ***Type*** (Register Type): Type one of the following (O=Output, I=Input, S=Status, B=Binary, N=Integer, T=Timer, C=Counter, R=Control, F=Float, A=ASCII, ST=String)
- ***SlotNumber***: Type the I/O card slot number
- ***TypeGroup***: Type the Group number of the register type configured
- ***OctetNumber****:* Type the Octet number of the I/O card configured in the **Header** field
- ***Format***: Type *(optional parameter)*
  - `W` to treat the values as words
  - `B` to treat the values as BCDs
  - `F` to treat the values as Floats (*double words*)
  - `S` to treat the values as Strings
- ***Address****:* Type the Address of the Group configured in the **Header** field.
- ***Number of Bytes****:* Specify the maximum size of ASCII/STRING data type.
- ***Bit****:* Type the bit number (from 0 up to 15) from the word address. (*optional parameter*).
- ***Element****:* Specify the element type for Timers, Counters, and Controls (refer to the table in the "Configuring the Address Field" section).

➲**Attention:** For worksheets that handles ASCII data type (**A** type in the Header and **S** type in the Address fields)
In PLC5 you'll be using "Typed Read/Write" commands. With this command you may read starting from an odd address, but you cannot write to an odd address. The final address is given by the sum of the initial address in the Header field and the least address found in the Address column of the worksheet. When writing ASCII data type, be sure to write to even addresses.

✍ **Note:** The String Format needs to have the size of the String. The size 2 is the size default.

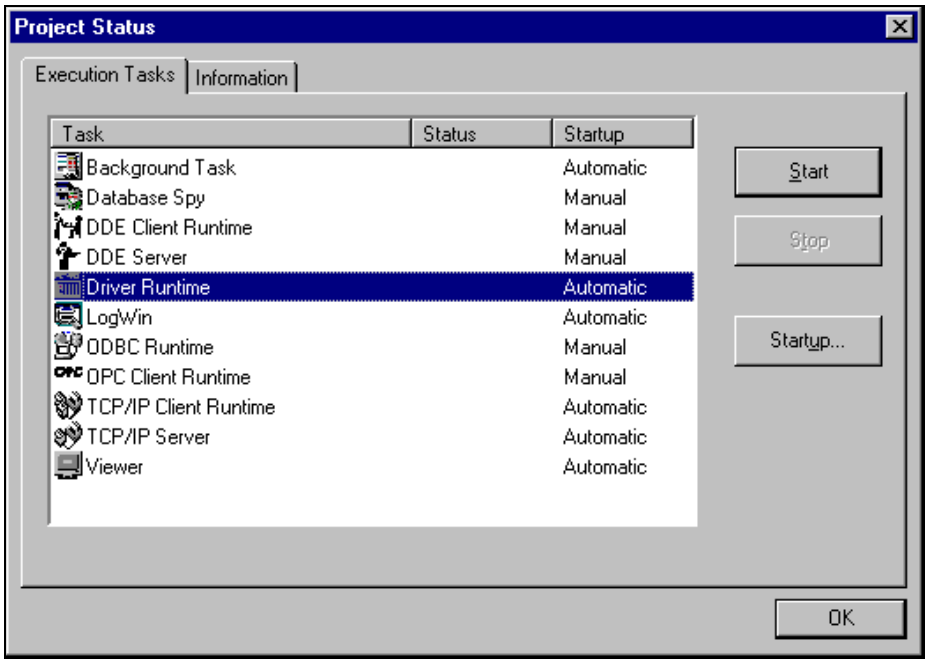
Executing the Driver

When you add the driver to a project, the system sets it automatically so the driver is ready to execute when you start-up the Runtime Environment.

To verify that the driver is enabled and will start correctly,

1. Select **Project** from the main menubar, and then select the **Status**... option from the menu to verify the Driver Runtime task.

The *Project Status* dialog box displays.
.



If you click on (highlight) the **Driver Runtime** line (as shown in the preceding figure), the **Startup**... button becomes active. You can click on the **Startup**... button to switch between Automatic and Manual Startup mode.

Troubleshooting

If the ABKE driver fails to communicate with the Allen Bradley device, the tag you configured for the **Read Status** or **Write Status** fields receives an error message. The error message contains an error code, which you can use to identify what type of failure occurred.

The following table describes the most common driver error codes. The error codes that have the prefix DF1: in the description are DF1 protocol error codes and the possible causes were extracted based on empirical tests.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required
-34	Invalid Address	Wrong Element specified for a Timer, Counter or Control parameter in the Address field. Wrong address syntax specified for other files	Check Address field.
-37	Invalid Header	The header specified is invalid	Check Header field. ▪ If you use a TAG, check whether the TAG value is valid for the specified addresses. ▪ if you did not use a TAG, you might have changed the Header after configuring the Addresses and the addresses are invalid for the new Header.
-39	Invalid block size	The worksheet exceeds the maximum address range supported by the protocol	Break your worksheet into several smaller worksheets.
7	DF1: File is wrong size	You are trying to access an element that exceeds the maximum range in a file.	Check your address range in the driver configuration and verify if it matches your declaration in the PLC.
10	DF1: Transaction size plus word address is too large	Requested block size is not supported by this CPU model	If you are using Standard Driver Sheet, break it in smaller ones
11	DF1: Access denied, improper privilege	A new program is being downloaded the PLC and the PLC model does not allow other communication while downloading. The PLC is protected by some internal mechanism which is not allowing the communication.	Check if a new program is being downloaded to the PLC. Check if the PLC is protected.
14	DF1: Command cannot be executed	The PLC family specified in the driver settings does not match the PLC that you are communicating with.	Check the PLC family in the driver settings.
17	DF1: Illegal data type	The file that you are reading does not exit in the PLC. The file type specified does not match the file type configured in the PLC.	Check if the file that you are trying to communicate with is properly configured in the PLC and if the file type matches.
27	DF1: Another node is the	A new program is being	Check if a new program is being downloaded

	program owner	downloaded the PLC and the PLC model does not allow other communication while downloading.	to the PLC.
256	Protocol Error	- The message received from the PLC has over 512 characters - The status code returned by the PLC is invalid - Trying to read a block larger than supported by this CPU in Family 500 PLC - Trying to read a block larger than supported by this CPU in Family 500 PLC - Trying to read an invalid file in a Family 500 PLC - Trying to read an invalid address from a valid file in a family 500 PLC - Trying to communicate with a MocroLogix CPU having the Wrong PLC family configured on the Communication Settings	Use the protocol analyzer in the LogWin to retrieve the STS error code and check the STS Error Codes table below.
257	Invalid address for this file type	ASCII file type of PLC5 requires the configuration of even addresses, if you specify an odd value you will get this error.	Change your address to be an even number

The following table lists all the error codes that can be returned by the DF1 protocol:

Error Code	Description
1	A field has an illegal value
2	Less levels specified in address than minimum for any address
3	More levels specified in address than system supports
4	Symbol not found
5	Symbol is of improper format
6	Address doesn't point to something usable
7	File is wrong size
8	Cannot complete request, situation has changed since the start of the command
9	Data or file is too large
10	Transaction size plus word address is too large
11	Access denied, improper privilege
12	Condition cannot be generated resource is not available
13	Condition already exists resource is already available
14	Command cannot be executed
15	Histogram overflow
16	No access
17	Illegal data type
18	Invalid parameter or invalid data
19	Address reference exists to deleted area

20	Command execution failure for unknown reason; possible PLC 3 histogram overflow
21	Data conversion error
22	Scanner not able to communicate with 1771 rack adapter
23	Type mismatch
24	1771 module response was not valid
25	Duplicated label
26	File is open; another node owns it
27	Another node is the program owner
28	Reserved
29	Reserved
30	Data table element protection violation
31	Temporary internal problem
34	Remote rack fault
35	Timeout
36	Unknown error

The following table lists the PLC STS errors when the driver returns *Protocol Error (Status 256)*

Error Code	Description
10	Illegal command or format.
20	Host has a problem and will not communicate
30	Remote node host is missing, disconnected, or shut down
40	Host could not complete function due to hardware fault
50	Addressing problem or memory protect rungs
60	Function not allowed due to command protection selection
70	Processor is in Program mode
80	Compatibility mode file missing or communication zone problem
90	Remote node cannot buffer command
A0	Wait ACK (1775 KA buffer full)
B0	Remote node problem due to download
C0	Wait ACK (1775 KA buffer full)
D0	Not used
E0	Not used
F0	Error code in the EXT STS byte
50	Addressing problem or memory protect rungs
60	Function not allowed due to command protection selection
70	Processor is in Program mode

Tip: You can verify the communication status using the Studio environment *Output* window or the *LogWin* module. To establish an event log for **Field Read Commands**, **Field Write Commands,** and **Serial Communication** right-click on the *Output* window. When the pop-up menu displays, select the option to set the log events. If you are testing under a Windows CE target, you can use the **Remote LogWin** (Menu *Tools->Remote LogWin)*

If you are unable to establish communication with the PLC, you must first try to establish communication between the PLC Programming Tool and the PLC. Quite frequently, communication is not possible because you have a hardware or cable problem, or a PLC configuration error. After you successfully establish communication between the PLC Programming Software and the PLC, you can retest this communication driver.

When testing communication with Studio, you should first try using the application sample in the next chapter (if the sample is available for the driver), instead of using the new application that you are creating.

If you are unable to establish communication between the ABKE driver and the Allen Bradley devices, you can contact the Technical Support staff as described in the *Studio Technical Reference Manual*.

Using the Application Sample

Studio provides a configured project that you can use to test the driver. We strongly recommend that you perform some tests with this application sample before configuring a customized project, for the following reasons:

- To better understand the information discussed in this document.
- To verify that your configuration is working.
- To certify that the hardware used in the test (device + adapter + cable + PC) is in working condition *before* you start configuring the application.

Note: The Application Sample is not available for all drivers.

You will find the Studio application sample on the installation CD-ROM in the following directory:

`\COMMUNICATION EXAMPLES\ABKE`

To perform the test, you must follow these steps:

1. Configure the device communication parameters using the manufacturer programmer software.
2. Open the `\COMMUNICATION EXAMPLES\ABKE` application.
3. Execute the application.
4. Display the application screen (which includes some information about the communication) by executing the Viewer module in Studio.

Tip: You can use the application sample as a maintenance screen for the custom application.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	2.04	Roberto V. Junior	Sep/30/1999	▪ First driver version ▪ Driver available for Windows CE
B	2.05	Roberto V. Junior	Jan/10/2000	Implemented SLC500 Bit Writing
C	2.06	Roberto V. Junior	Apr/7/2000	Fixed “Timeout Between Rx char” error caused by bit write before first Read command.
D	2.07	Roberto V. Junior	Jul/7/2000	▪ Write error when first address position is not zero. ▪ Write error when addresses are out of sequence.
E	2.08	Roberto V. Junior	Aug/2/2000	▪ Added CRC Check Error ▪ Fixed bug in the Byte counter (timeout error)
F	2.09	Roberto V. Junior	Aug/22/2000	Show Rx message in the LogWin.
G	2.10	Roberto V. Junior	Oct/9/2000	Added MAIN DRIVER SHEET feature.
H	2.11	Roberto V. Junior	Jan/1/2001	Added Timer, Counter, and Control operands.
I	2.12	Roberto V. Junior	Mar/20/2001	Fixed configuration of I and O operand in the MAIN DRIVER SHEET.
J	2.13	Roberto V. Junior	Mar/23/2001	Fixed problem caused by reading Timers with offsets greater than 0.
K	2.14	Roberto V. Junior	Jun/26/2001	▪ Added Float-Pointer operand. ▪ Fixed bug to wait device response.
L	2.15	Roberto V. Junior	Jul/20/2001	Added ASCII operand.
M	2.16	Roberto V. Junior	Jul/24/2001	Added String operand.
N	2.16	Fabíola Fantinato	Nov/20/2001	Added Appendix A to the documentation.
O	2.17	Roberto V. Junior	Feb/21/2002	▪ Documentation format and text revision. ▪ Included “Typed Read/Write” commands to PLC5 family
P	2.18	Eric Vigiani	Jun/5/2002	Modified internal algorithm to accept initial address higher than 255.
Q	2.20	Eric Vigiani	Sep/30/2002	Modified header parse to accept any Type Group of the PLC 5 family
R	2.21	Lourenço Teodoro	Aug/11/2003	Fixed GPF when the address configured in the Main Driver Sheet is invalid.
S	2.22	Fabio H. Y. Komura	Feb/20/2004	Fixed error when configure several Float-Point data registers in the Main Driver Sheet.
T	2.23	Fabio H. Y. Komura	Jul/15/2004	▪ Fixed problem with I/Os in family 5. ▪ Included the option to choose the address for data type as Octal or Decimal to PLC5. ▪ Fixed problem when “Write with Header” offset is different from 0 (zero).
U	2.23	Lourenço Teodoro	Nov/23/2004	Updated the communication parameters window.
V	2.24	Eric Vigiani	Apr/13/2005	Fixed problem in the read to PLC2 family
W	2.25	Diego Barros	Apr/17/2005	▪ Fixed problem while reading Float operands. MDS and Standard sheets. ▪ Fixed problem on MDS – Timer and Counter Block Size. ▪ Fixed problem while writing strings to both MDS and SDS.

X	2.28	Plínio M. Santana	Mar/26/2007	▪ Fixed problems on reading and writing operations for ASCII type (type String on Address). ▪ Fixed problem to allow the tag's size be larger than 2 bytes. ▪ Inserted information about Default Format for Address field. ▪ All of address validation and the register types corrected. ▪ String tags working for more operands. ▪ Operation with bits working for ASCII operands. ▪ Document modification: ➢ Accept address without format. ➢ Registers table updated. ➢ Sample of Address configuration table (BCD column). ➢ Included 1234 Error code. ➢ Warn about the String size Default.
Y	2.29	Plínio M. Santana	Ago/01/2007	▪ Fixed timeout problems on Windows CE.
Z	2.30	Eric Vigiani	Feb/08/2008	▪ Modified the PLC Family options. ▪ Fixed problem with ASCII operands. ▪ Modified to work with the encapsulation feature.
AA	10.1	Marcelo Carvalho	Jan/07/2009	▪ Updated driver version, no changes in the contents.
AB	10.2	Joel Nascimento	Apr/06/2009	▪ Implemented Legacy/Standard BCD opitions.
AC	10.3	Lourenço Teodoro	Jun/26/2009	▪ Reviewed errors section.
AD	10.4	Paulo Balbino	Jan/23/2013	▪ Fixed crash problems.
AE	10.5	Anushree Phanse	Apr/23/2015	▪ Fixed timeout issue on WinCE devices
AF	10.6	Anushree Phanse	Nov/3/2016	▪ Fixed the issue when reading the last address in a floating point register

Appendix: Setting Up PLC Communication Using Rockwell RSLogix Software

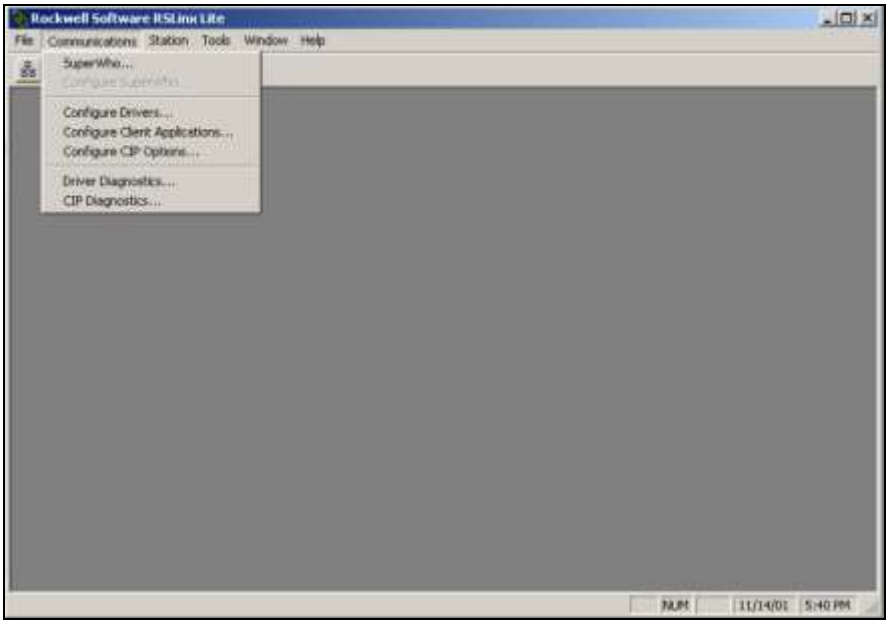
Rockwell RSLogix software is a workbench tool that you must use with RSLinx software to enable communication with the PLC.

✍ **Note:** Before attempting to establish communication with Studio, we recommend using the RSLogix software to set up communication with a serial line to ensure that the PLC, connections, and cable are all working properly.

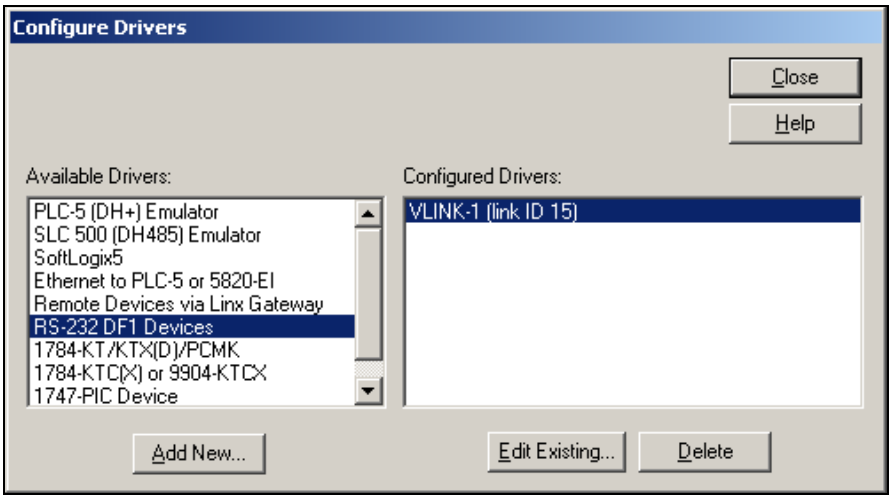
Use the following step-by-step procedure to set up communication with RSLogix using a serial line.

✍ **Note:** These instructions use *PLC* to refer to a SLC500 or a KE module. You must configure the device in the appropriate field.

1. Connect your computer and the PLC through a serial cable.
2. Start the RSLinx software.
3. Select **Communications** from the *Rockwell Software RSLinx Lite* menubar, and then select **Configure Drivers** from the menu (as shown in the following figure).



4. When the *Configure Drivers* dialog displays, select **RS-232 DF1 Devices** from the **Available Drivers** list, and click the **Add New**... button.

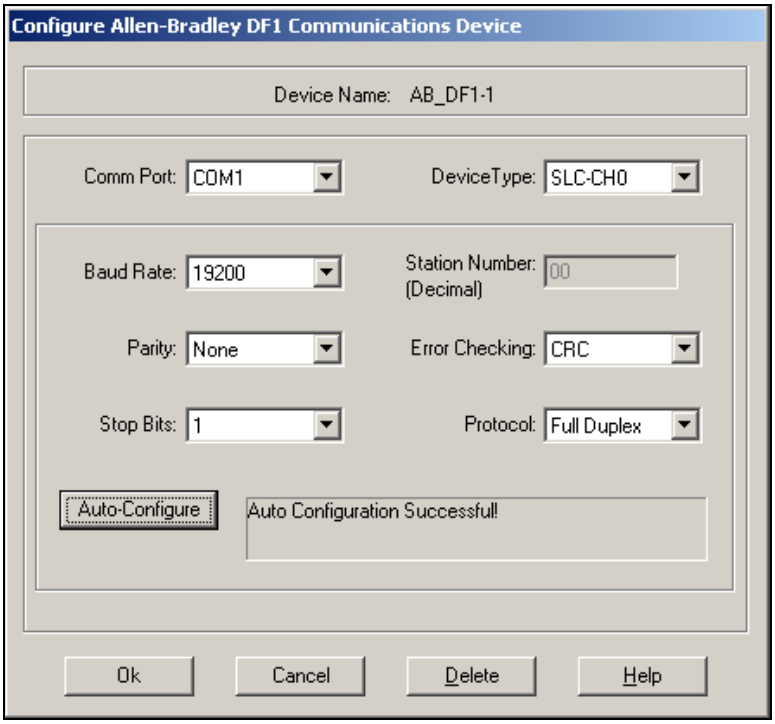


5. The **RS-232 DF1 Devices** driver moves to the **Configured Drivers** list. Click the **Close** button to save your changes and close the dialog box.
6. When the *Configure Allen-Bradley DF1 Communications Device* dialog displays, specify an appropriate **Comm Port** and **Device Type**.
7. You can configure the remaining parameters on this dialog automatically by clicking the **Auto-Configure** button.

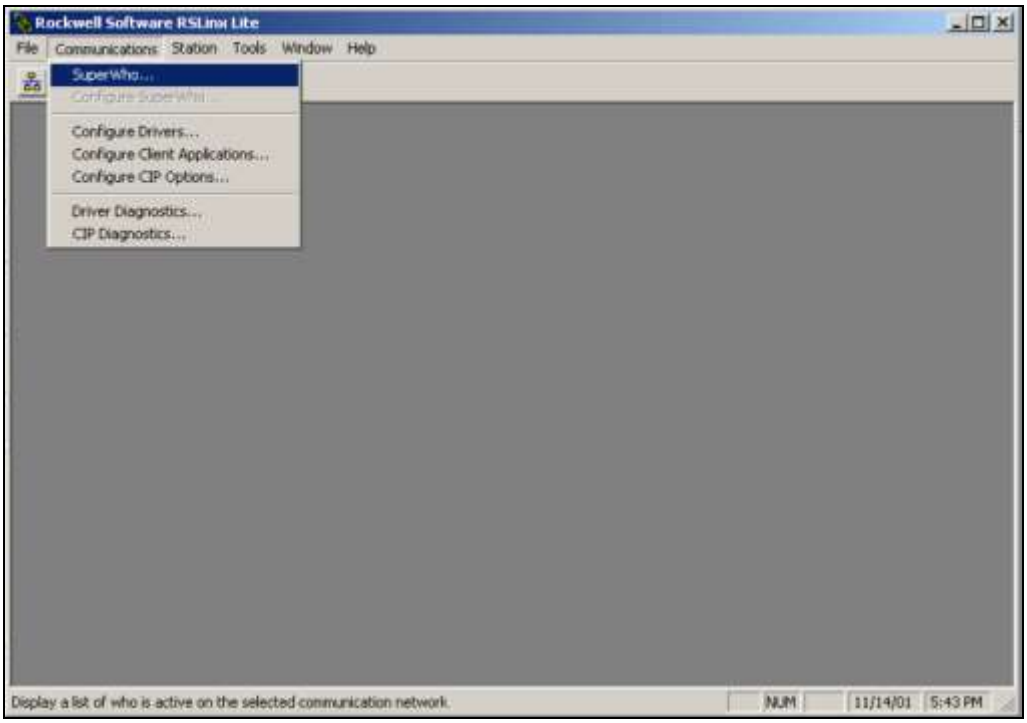
After a few moments, you should see the following message:

```
Auto Configuration Successful!
```

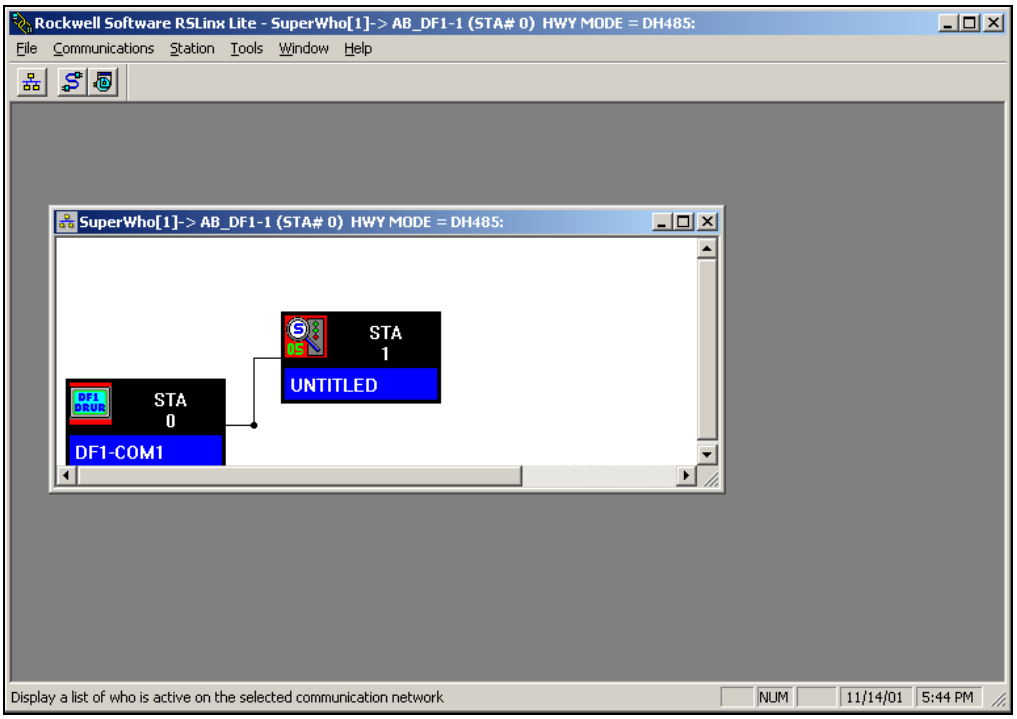
If this message does not display, verify that the **Comm Port** and **Device Type** parameters are correct and then check that the cable is working properly.



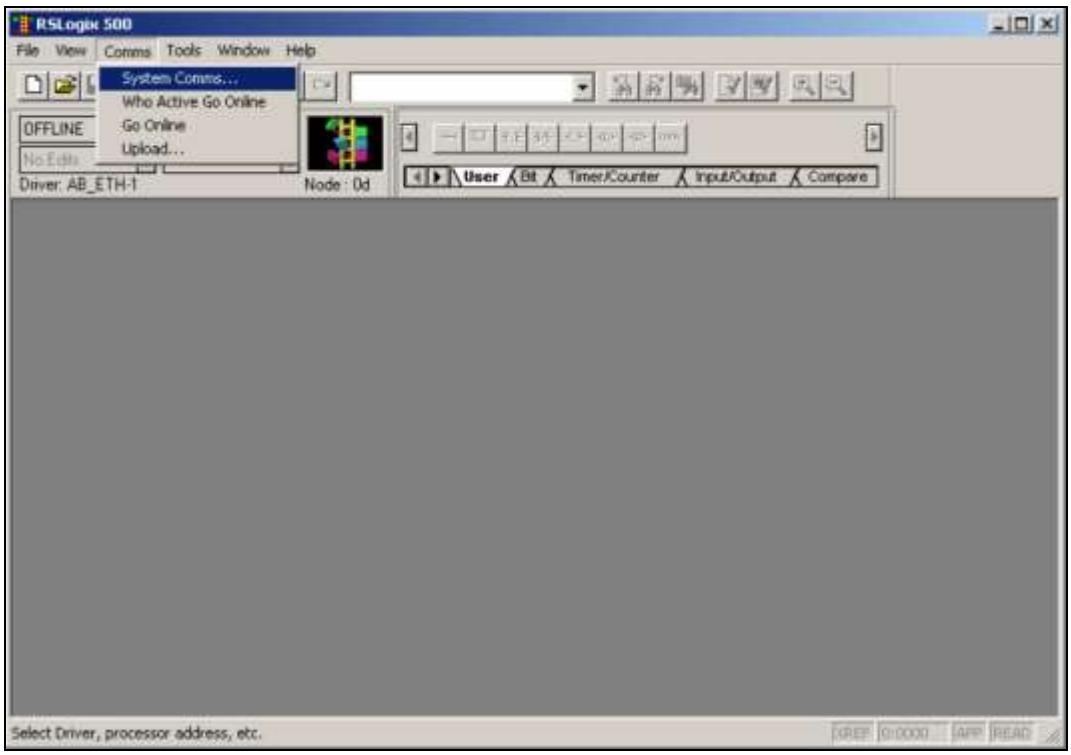
8. Click **Ok** to confirm your settings and close the dialog box.
9. Return to the main menubar. Select **Communications** and then select the **SuperWho**... option.



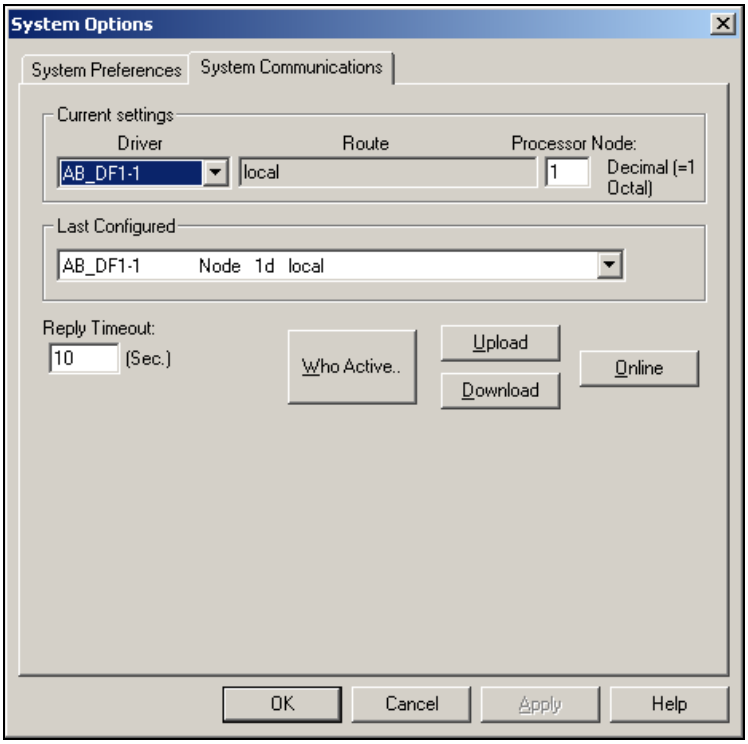
You should see the following window, which represents the PC Station 0 and the PLC Station 1 communicating.



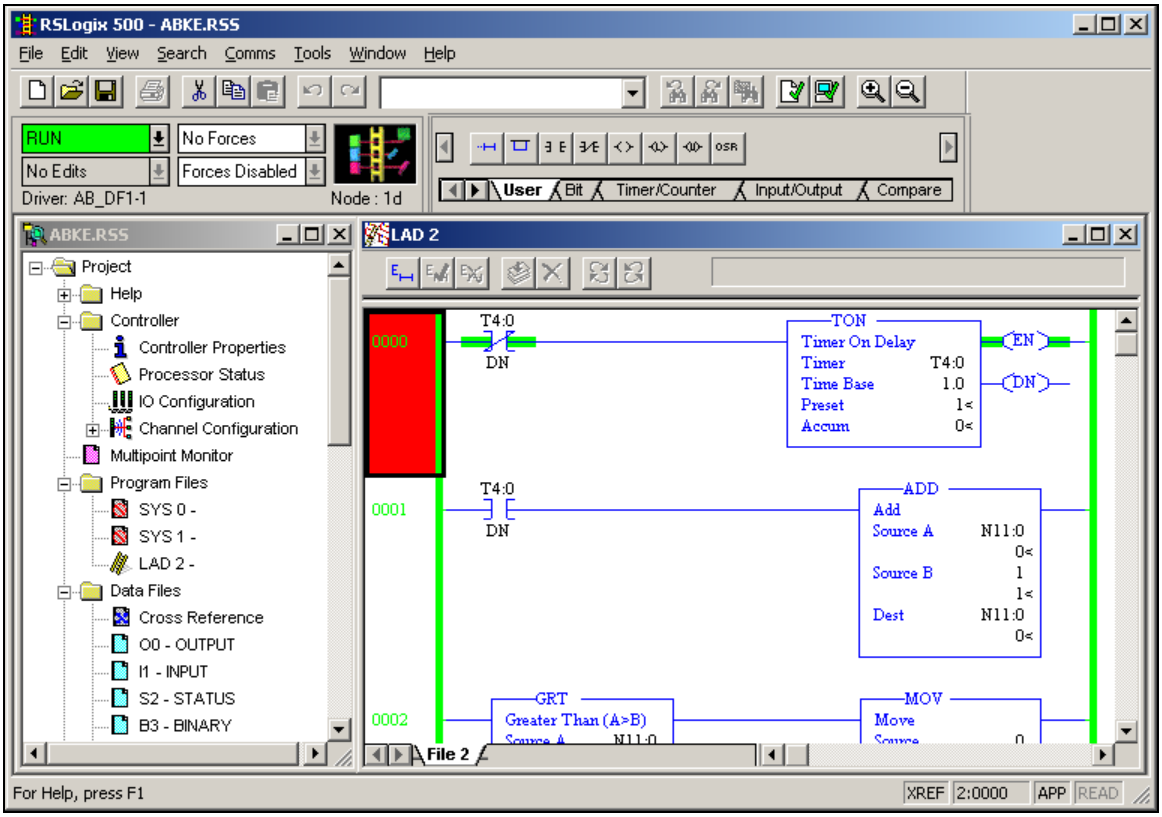
10. Start the **RSLogix** software.
11. From the *RSLogix* main menubar, select the **Comms** option and then select **System Comms**....

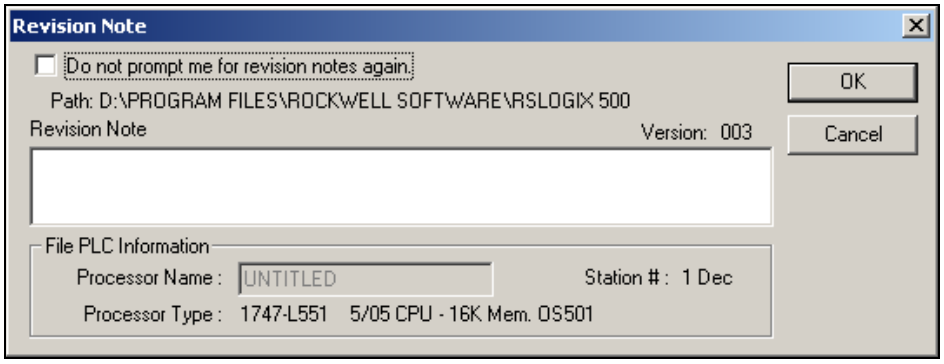


12. When the *System Options* dialog displays, click the **System Communications** tab.
13. In the **Current settings** section, select the AB_DF1-1 driver from the **Driver** drop-down list.
14. In the **Processor Node** text box, type the PLC station number: `1`.
15. Click **OK** to close this dialog.



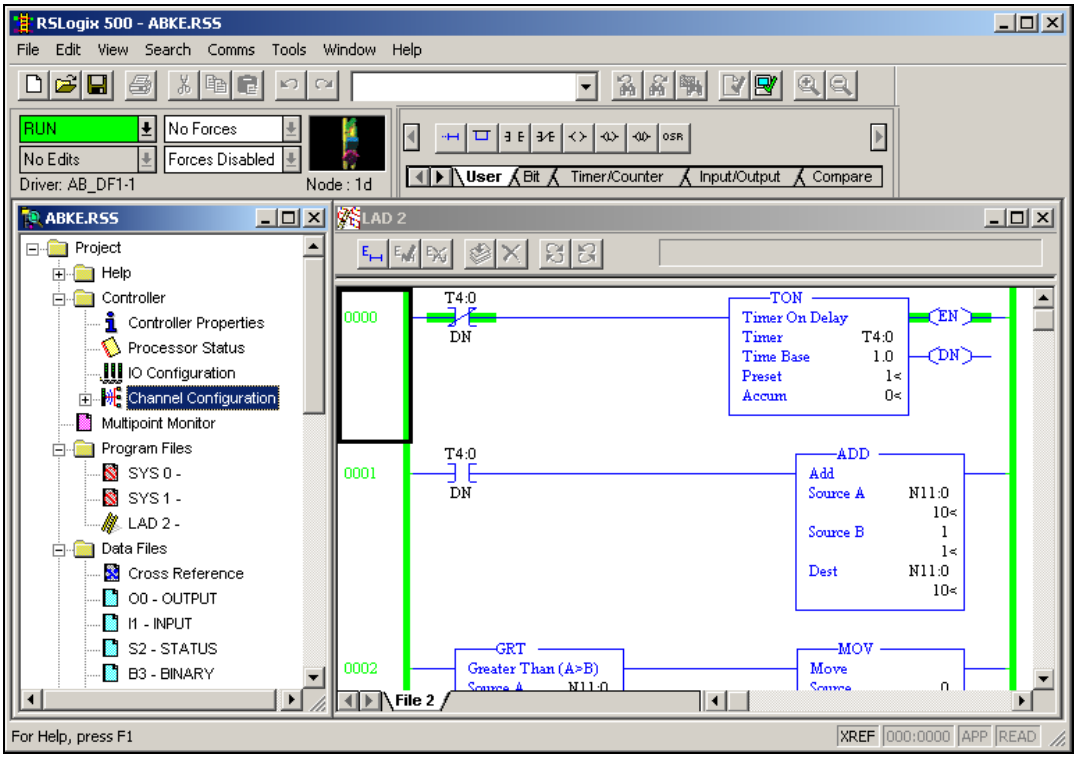
You should see the following screen in the *RSLogix 500* window and the *Revision Note* dialog box:





16. Select **Comms** from the main menubar, and then select **Go Online**.

    If the RSLogix software is communicating successfully with the field device, you should see the following screen:



At this point, you can start testing communication with the Studio software.