

ABCIP Communication Driver

Driver for TCP/IP Ethernet Communication
with Devices Using the ABCIP Protocol

Contents

INTRODUCTION2

GENERAL INFORMATION.....3

 DEVICE CHARACTERISTICS3

 LINK CHARACTERISTICS.....3

 DRIVER CHARACTERISTICS3

 CONFORMANCE TESTING4

INSTALLING THE DRIVER5

CONFIGURING THE DEVICE6

CONFIGURING THE DRIVER6

 SETTING THE COMMUNICATION PARAMETERS6

 CONFIGURING THE STANDARD DRIVER WORKSHEET6

 CONFIGURING THE MAIN DRIVER WORKSHEET9

EXECUTING THE DRIVER11

TROUBLESHOOTING12

SAMPLE APPLICATION14

REVISION HISTORY.....15

Introduction

The ABCIP driver enables communication between the Studio system and the Allen Bradley ControlLogix PLC, FlexLogix and CompactLogix via 1756-ENET Board according to the specifications discussed in this document.

The ABCIP driver supports also routing communication for two levels. Therefore, using this driver, you can exchange data with a ControlLogix PLC directly connected in the rack where the 1756-ENET module is connected, or you can address other PLCs connected to the ControlLogix rack via:

- 1756-ENET: Communication interface for Ethernet/IP protocol.
- 1756-DHRIO: Communication interface for DH+ or Remote I/O (RIO).

This document was designed to help you install, configure and execute the ABCIP driver to enable communication with these ABCIP devices. The information in this document is organized as follows:

- **Introduction:** Provides an overview of the ABCIP driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the ABCIP driver.
- **Installing the Driver:** Explains how to install the ABCIP driver.
- **Configuring the Device:** Explains how to configure the ABCIP device.
- **Configuring the Driver:** Explains how to configure the communication driver.
- **Executing the Driver:** Explains how to execute the driver to verify that you installed and configured the driver correctly.
- **Troubleshooting:** Lists the most common error codes for this protocol and explains how to fix these errors.
- **Sample Application:** Explains how to use a sample application to test the driver configuration.
- **Revision History:** Provides a log of all modifications made to the driver and the documentation.
- **Appendix:** Families 5, 500 Routing Capabilities and MicroLogix 1100

Notes:

- This document assumes that you have read the “Development Environment” chapter in the product’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Windows NT/2000/XP environment. If you are unfamiliar with Windows NT/2000/XP, we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter explains how to identify all the hardware and software components used to implement serial communication between the Studio ABCIP driver and devices using the ABCIP protocol.

The information is organized into the following sections:

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

Device Characteristics

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

- **Manufacturer:** Allen Bradley/Rockwell
- **Compatible Equipment:**
 - ControlLogix 5550 series, 1756-ENET board
 - FlexLogix
 - CompactLogix
 - Micrologix 1100
- **PLC Programming Software:** Rockwell RSLogix5000 and Rockwell RSLogix500

For a list of the devices used for conformance testing, see “Conformance Testing” on page 4.

Link Characteristics

To establish serial communication, you must use links with the following specifications:

- **Device Communication Port:** Ethernet Port on the 1756 ENET Module
- **Physical Protocol:** Ethernet
- **Logic Protocol:** CIP over Ethernet/IP
- **Device Runtime Software:** None
- **Specific PC Board:** Ethernet Port
- **Cable Wiring Scheme:** Regular Ethernet Cable

Driver Characteristics

The ABCIP driver is composed of the following files:

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

Notes:

- All of the preceding files are installed in the /DRV subdirectory of the Studio installation directory.
- You must use Adobe Acrobat® Reader™ (provided on the Studio installation CD-ROM) to view the **ABCIP.PDF** document.

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

- Windows 2000/XP
- Windows NT
- Windows CE 3.0 and Windows CE.Net

For a list of the operating systems used for conformance testing, see the “Conformance” section that follows.

The ABCIP driver supports the following address types:

| Register Type | Length | Write | Read | Bit |
|---------------|--------------|-------|------|-----|
| BOOLEAN | 1 Bit | • | • | – |
| INT | 2 Bytes | • | • | • |
| SINT | 2 Bytes | • | • | • |
| DINT | 4 Bytes | • | • | • |
| REAL | 4 Bytes | • | • | – |
| STRING | Configurable | • | • | – |

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration:**

- PLC Program: N/A
- Modbus Port: N/A
- Baud Rate: N/A
- Protocol: N/A
- Data Bits: N/A
- Stop Bits: N/A
- Parity: N/A
- COM Port: N/A

- **Cable:** Regular Ethernet cables and HUBs

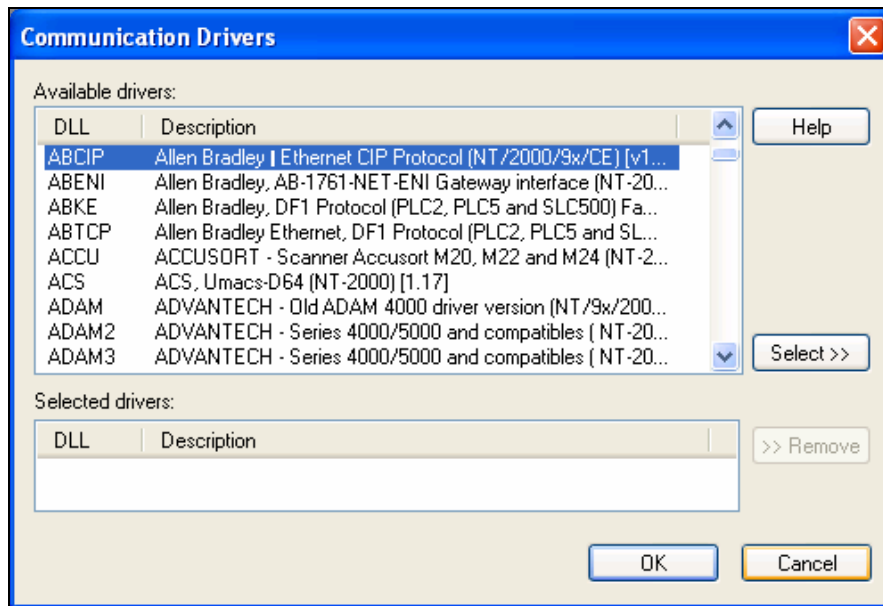
| Driver Version | Studio Version | Operating System (development) | Operating System (target) | Equipment |
|----------------|----------------|--------------------------------|--------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.09 | 6.1 | Windows XP/2000 | <ul style="list-style-type: none"> ▪ Windows NT 4.0/2000/XP ▪ Windows CE | <ul style="list-style-type: none"> • PLC Allen Bradley 1756-L1 ControlLogix 5550 Controller + 1756ENET/B Module • PLC Allen Bradley MicroLogix 1100 1763-L16BWA • PLC5/80 and SLC5/05 via routing |

Installing the Driver

When you install Studio version 5.1 or higher, all of the communication drivers are installed automatically. You must select the driver that is appropriate for the application you are using.

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

1. Open Studio from the **Start** menu.
2. From the Studio main menu bar, select **File** → **Open Project** to open your application.
3. Select **Insert** → **Driver** from the main menu bar to open the *Communication Drivers* dialog.
4. Select the **ABCIP** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

5. When the **ABCIP** driver displays in the **Selected Drivers** list, click the **OK** button to close the dialog.

Note:

It is not necessary to install any other software on your computer to enable communication between Studio and the device. However, to download the custom program to your device, you must install a RSLogix5000 package. Consult the Rockwell documentation for installation instructions.

Caution:

For safety reasons, you must use special precautions when installing the physical hardware. Consult the hardware manufacturer's documentation for specific instructions in this area.

Configuring the Device

Configure a valid IP address to your 1756-ENET board, and place PLC in RUN mode. You might already be able to communicate with it.

Configuring the Driver

After opening Studio and selecting the ABCIP driver, you must configure the driver. Configuring the ABCIP driver is done in two parts:

- Specifying communication parameters
- Defining communication tags and controls in the Communication tables or *Driver* worksheet

Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header** and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header** and **Address** fields only.

Notes:

For information about how to the ABCIP driver to communicate with **MicroLogix 1100** and with PLC5 and SLC500 families via **Routing**, please see the **Appendix** in the end of this document

For a detailed description of the Studio Standard and Main Driver Worksheets, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

Setting the Communication Parameters

Note:

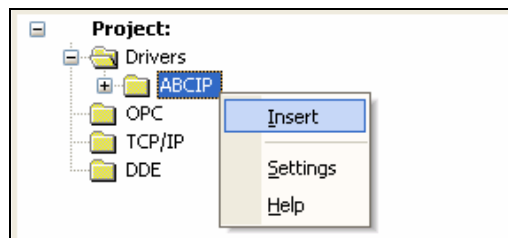
You do not need to adjust the configuration in the *ABCIP Communication Parameters* window.

Configuring the Standard Driver Worksheet

This section explains how to configure a *Standard Driver Worksheet* (or Communication table) to associate application tags with the PLC addresses. You can configure multiple *Driver* worksheets—each of which is divided into a *Header* section and a *Body* section.

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

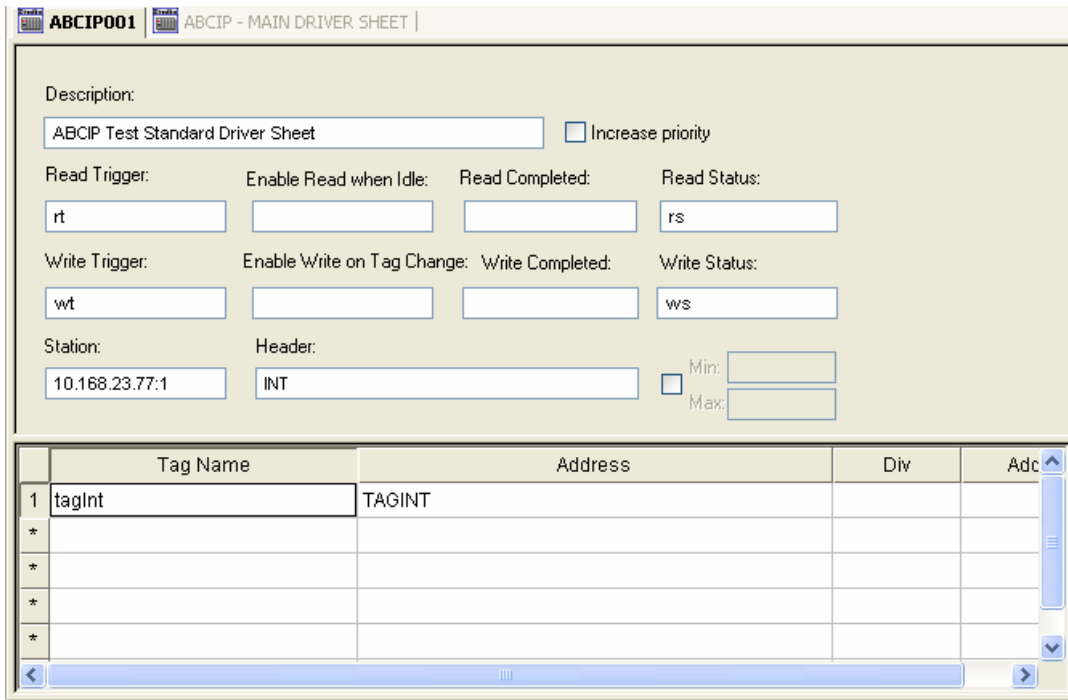
1. From the Studio development environment, select the **Comm** tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the *ABCIP* subfolder.
3. When the pop-up menu displays, select the **Insert** option:



Inserting a New Worksheet

Note:
 To optimize communication and ensure better system performance, you must tie the tags in different driver worksheets to the events that trigger communication between each tag group and the period in which each tag group must be read or written. Also, we recommend configuring the communication addresses in sequential blocks to improve performance.

The ABCIP.drv dialog displays (similar to the following figure):



ABCIP Driver Worksheet

4. Use the following information to complete the **Station**, **Header** and **Address** fields on this worksheet.
 - **Station** field: Use this field to specify the IP Address of the device (*unit name*) and the slot number, using the following syntax:
 - For ControlLogix, FlexLogix and CompactLogix
<IP Address>:<Slot>
 - For MicroLogix 1100
1100:<IP Address>

Note:
 This driver supports routing to other PLCs from other families (5 and 500). Please check the Appendix in the end of this document for further information about it.

- **Header** field: Use the information in the following table to define the type of variables that will be read from or written to the device and a reference to the initial address.

These variables must comply with the following syntax:

<**Type**> (for example: INT) or

<**Type**><[**Array Dim**]> (for example: INT [k, m, n]) where k, m and n are the higher array dimensions.

Where:

<**Type**> is the PLC Tag type. Use one of the following: INT, SINT, DINT, BOOL, REAL or STRING

After you edit the **Header** parameter, the system checks that the syntax is valid. If the syntax is invalid, Studio automatically inserts the default value (*DINT*) into the **Header** field.

Also, you can type a tag string in brackets {**Tag**} into the **Header** field, but you must be certain that the tag's value is correct and that you are using the correct syntax, or you will get an invalid Header error.

The following table lists all of the data types and address ranges that are valid for the **Header** field. Notice that these are basically the *Atomic* Data Types.

| Data Types | Sample Syntax | Comments |
|------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INT | INT | INT: Reads and writes the INT data type, which consists of a WORD 16 bits signed Integer Value (-32768 to 32767). Max Block Size: 240 elements |
| DINT | DINT | DINT: Reads and writes the DINT data type, which consists of a DWORD 32 bits signed Integer Value (-2,147,483,648 to -2,147,483,647). Max Block Size: 120 elements |
| SINT | SINT | SINT: Reads and writes the SINT data type, which consists of a Byte 8 bits signed Integer Value (-128 to 127). Max Block Size: 480 elements |
| REAL | REAL | REAL: Reads and writes the REAL data type, which consists of a 32Bits-IEEE Floating Point Value (-128 to 127). Max Block Size: 120 elements |
| BOOL | BOOL | BOOL: Reads and writes the BOOL data type, which consists of a 1 bit value (0 or 1) |
| STRING | STRING | STRING: Reads and writes the STRING data type, which stores up to 82 characters |

- **Address** field: Type the Tag Name.

Type the tag from your application database into the **Tag Name** column. This tag will receive values from or send values to an address on the device. The address must comply with the following syntax:

- <**Logix Tag Name**>(for example, ControlNet123) or
- <**Data Type**>:<**Logix Tag Name** [k, m]> (for example, DINT:Control [1, 2]) or
- <**StringTagName**>.DATA [0] (for example: STRING:STR.DATA [0])

Where:

<**Logix Tag Name**> is the Tag Name used on the RSLogix Tag Database.

➔ **Cautions:**

- You can use the Bit Writing function only with the *Write on tag change* driver tag enabled, which means that you cannot use the *Write trigger tag* for the Bit Writing function.
- You must not configure a range of addresses greater than the maximum block size (*data buffer length*) supported by each PLC within the same *Driver* worksheet. The maximum data buffer length for this driver is shown in the previous table, in the *Comments* column for each data type.

To Program Tags (tags outside the controller tags database), you need to type the following:

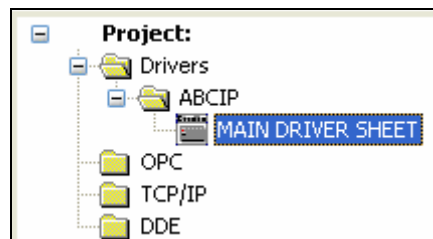
PROGRAM: <ProgramName> . <Logix Tag Name>

For example: **PROGRAM:MainProgram.UDT1.M1 [0]**

| Address Configuration Sample | | |
|------------------------------|--------------|-------------------------------------|
| Device Address | Header Field | Address Field |
| INT[0] | INT | INT[0] |
| INT[0]/3 (bit 3) | INT | INT[0]/3 |
| Control[1,2,0] | INT[3,3,3] | Control[1,2,0] |
| Control123/5 (bit 5) | INT | Control123/5 |
| REAL[12] | REAL | REAL[12] |
| TEXTTEST | STRING | TEXTTEST.DATA:20 |
| B3[160] | BOOL | B3[160] |
| ROW123 | DINT | ROW123 |
| ROW123/2 (bit 2) | DINT | ROW123/2 |
| T4[0].ACC | DINT | DINT:T4[0].ACC |
| C5[10].PRE | DINT | DINT:C5[10].PRE |
| C5[10].PRE (Program Tag) | DINT | DINT:PROGRAM:MainProgram.C5[10].PRE |
| TestString | STRING | PROGRAM:MainProgram.TestString[0] |

Configuring the MAIN Driver Worksheet*

When you add the ABCIP driver to your application, the program automatically adds the *MAIN Driver Worksheet (MDS)* to the *ABCIP* driver folder as shown:



Main Driver Sheet Icon

You can use the MDS to associate Studio tags to addresses in the PLC. Most MDS parameters are standard for any driver, and are not discussed in this publication. For information about configuring these parameters, consult the *Studio Technical Reference Manual*.

* These Worksheets have not been fully tested with the current version of this driver.

Use the following instructions to configure the parameters that are specific to the ABCIP driver:

1. Double-click on the *Main Driver Sheet* icon to open the following worksheet:

| | Tag Name | Station | I/O Address | Action | Scan | Div | Add |
|---|---------------|----------------|-----------------------------------------------|------------|--------|-----|-----|
| 1 | tagString | 10.168.23.77:1 | STRING:PROGRAM:MAINPROGRAM.LOCAL.DATA[0] | Read+Write | Always | | |
| 2 | tagIntTest | 10.168.23.77:1 | INT:PROGRAM:MAINPROGRAM.INTTEST | Read+Write | Always | | |
| 3 | tagStringTest | 10.168.23.77:1 | STRING:PROGRAM:TESTPROGRAM.TESTSTRING.DATA[0] | Read+Write | Always | | |
| * | | | | Read+Write | Always | | |
| * | | | | Read+Write | Always | | |
| * | | | | Read+Write | Always | | |
| * | | | | Read+Write | Always | | |
| * | | | | Read+Write | Always | | |

Main Driver Sheet

2. Configure the following fields on this worksheet:

- **Station field:** Type the PLC IP address and the Slot Number, using the following syntax:
 <IP Address>:<Slot>

Caution:
 You must use a non-zero value in the **Station** field, and you cannot leave the field blank.

- **I/O Address field:** Type the tag name of each PLC address, using the following syntax:
 <Type>:<PLC Tag Name> (for example, INT:NT[0], INT:Row[2], BOOL:B3[34], REAL:Temp[127], INT:Var[1,2]) or **STRING:String Tag Name**.DATA[0] (for example, STRING:TEXT.DATA[0], STRING[2,4,3]:TEXT[1,3,2].DATA[0])

Where:

- <Type> is the register type. Use one of the following: **BOOL**, **INT**, **SINT**, **DINT**, **REAL** or **STRING** plus **higher array dimensions** (if exists)
- <PLC Tag Name> is the PLC Tag Name (address).

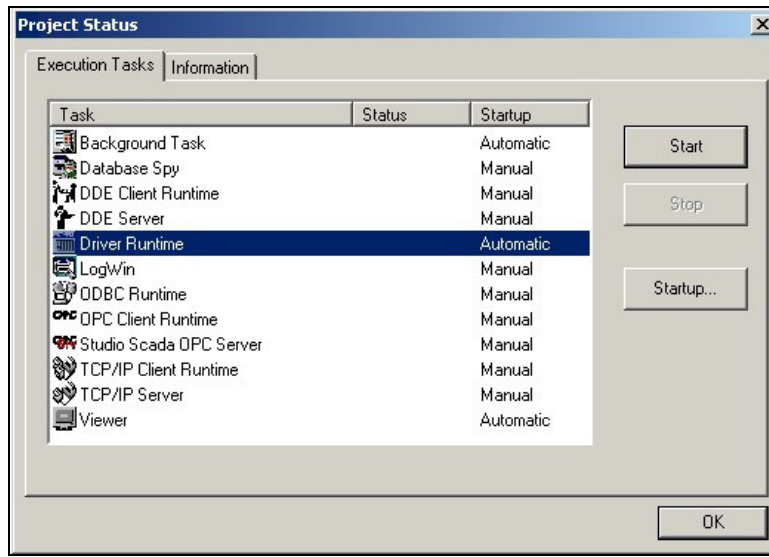
Executing the Driver

After adding the ABCIP driver to a project, Studio sets the project to execute the driver automatically when you start the run-time environment.

To verify that the driver run-time task is enabled and will start correctly, perform the following steps:

1. Select **Project** → **Status** from the main menu bar.

The *Project Status* dialog displays:



Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
 - If the setting is correct, click **OK** to close the dialog.
 - If the **Driver Runtime** task is set to **Manual**, select the **Driver Runtime** line. When the **Startup** button becomes active, click the button to toggle the *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the ABCIP driver fails to communicate with the device, the tag you configured for the **Read Status** or **Write Status** fields 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 required |
| 3 | Invalid Command | <ul style="list-style-type: none"> Trying to write BOOL Data Type using the "Write Trigger" driver worksheet tag Wrong Data Type in the Driver Worksheet Header field | <ul style="list-style-type: none"> The Tags of Data Type BOOL can only be written via "Enable Write on tag change" or Main Driver Sheet. Type a valid header (INT, SINT, DINT, REAL, BOOL, STRING). |
| 7 | Invalid Block Size | More items than allowed in a Standard Driver Worksheet | Split your driver worksheet into two or more. See the limits for each data type in the table (<i>Comments</i> column) on page 8. |
| 8 | Invalid Write Command | The PLC didn't recognize the tag that the application is trying to write. | <ul style="list-style-type: none"> Check the Tag Data Type in your driver worksheet. Check the Tag Name in your driver worksheet. Check the Address Syntax in your driver worksheet. |
| 10 | Not Allocated Memory | The driver is trying to remove memory that was not previously allocated. | This is a driver internal error. If this error persists, please contact technical support. |
| 11 | Invalid Read Command | The PLC didn't recognize the tag that the application is trying to read. | <ul style="list-style-type: none"> Check the Tag Data Type in your driver worksheet. Check the Tag Name in your driver worksheet. Check the Address Syntax in your driver worksheet. |
| 22 | Invalid Data Type | The data type specified in the Driver Worksheet Address Field is not a valid one. | Type one of the valid Types (INT, DINT, SINT, REAL, BOOL, STRING). |
| 23 | Error in send_RR_data Function | The driver is not getting the logical connection to the PLC. | This is a driver internal error. If this error persists, please contact technical support. |
| 1005 | TimeOut | The SLOT configuration in STATION field may be incorrect. | Correct the slot configuration. |
| -15 | Timeout Start Message | <ul style="list-style-type: none"> Disconnected cables PLC is turned off, in stop mode, or in error mode Wrong station number | <ul style="list-style-type: none"> Check cable wiring. Check the PLC state – it must be RUN. Check the station address. Try to "ping" your PLC. |
| -17 | Timeout Between rx char | <ul style="list-style-type: none"> PLC in stop mode or in error mode Wrong station number | <ul style="list-style-type: none"> Check the PLC state – it must be RUN. Check the station address. |

⇒ **Tip:**

You can verify communication status using the Studio development environment *Output* window (*LogWin* module). To establish an event log for **Field Read Commands**, **Field Write Commands** and **Serial Communication**, right-click in the *Output* window. When the pop-up menu displays, select the option to set the log events. If you are testing a Windows CE target, you can enable the log at the unit (**Tools** → **LogWin**) and verify the **celog.txt** file created at the target unit.

If you are unable to establish communication with the PLC, 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 successfully establishing communication between the device's Programming Tool and the PLC, you can retest the supervisory driver.

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.

Sample Application

This driver does not have a sample application.

Revision History

| Doc. Revision | Driver Version | Author | Date | Description of Changes |
|---------------|----------------|------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A | 1.01 | Eric Vigiani | Oct/03/2003 | Initial version |
| B | 1.02 | Eric Vigiani | Jun/28/2004 | <ul style="list-style-type: none">Modified the driver to accept multiple connectionsModified driver must support communication with remote PLCs (SLC, PLC5, ControlLogix) via AB-1756 gatewayFixed bug to read boolean data properly |
| C | 1.03 | Fabio H.Y Komura | Jul/19/2004 | <ul style="list-style-type: none">Implemented read/write of bits |
| D | 1.04 | Fabio H.Y Komura | Oct/29/2004 | <ul style="list-style-type: none">Fixed bug with STRING headerFixed bug to get Header information properly |
| E | 1.05 | Leandro Coeli | Jan/27/2005 | <ul style="list-style-type: none">Fixed problems in MDSImplemented Multi-Array |
| F | 1.06 | Leandro Coeli | Mar/27/2005 | <ul style="list-style-type: none">Fixed problems in Station field |
| G | 1.07 | Leandro Coeli | Sep/12/2005 | <ul style="list-style-type: none">Fixed problems in Station field |
| H | 1.08 | Leandro Coeli | Jan/26/2006 | <ul style="list-style-type: none">Modified driver to accept the Program Operand |
| I | 1.09 | Diego Barros | Apr/13/2006 | <ul style="list-style-type: none">Modified driver to communicate with MicroLogix1100 |

Appendix – Families 5, 500 Routing Capabilities and MicroLogix 1100

The ABCIP supports routing communication for two levels. Therefore, using this driver, you can exchange data with a ControlLogix PLC directly connected in the rack where the 1756-ENET module is connected, or you can address other PLCs connected to the ControlLogix rack via:

- 1756-ENET: Communication interface for Ethernet/IP protocol.
- 1756-DHRIO: Communication interface for DH+ or Remote I/O (RIO).
- 1756-CNB: Communication interface for ControlNet.

The typical configuration allows IWS to exchange data with the ControlLogix CPU directly connected where the 1756-ENET interface is connected, or with a Micrologix 1100, as illustrated in the following pictures

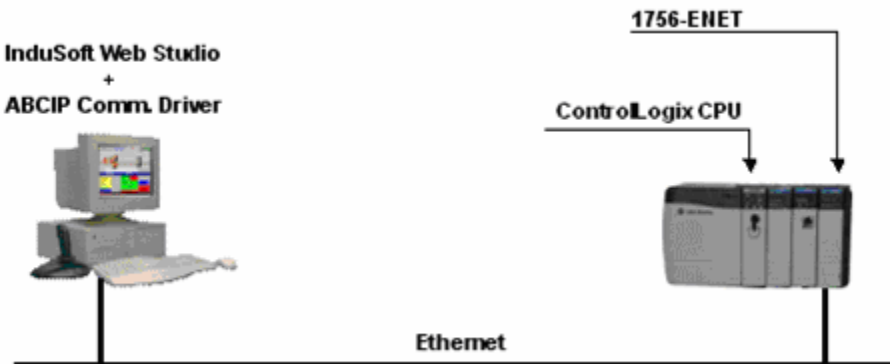
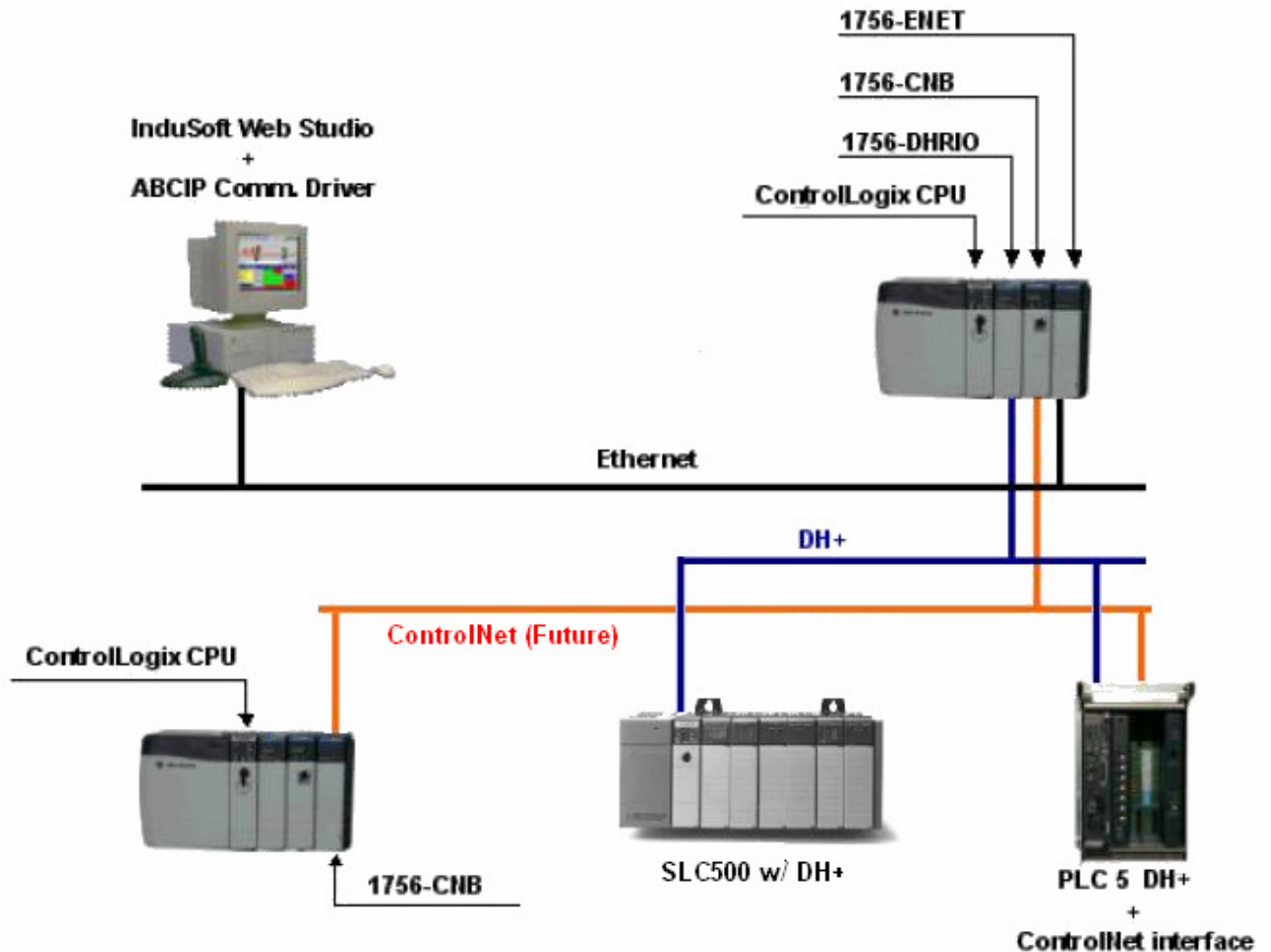


Fig: Communication with ControlLogix 55XX CPU



Fig: Communication with Micrologix 1100

The following picture illustrates the capability of the ABCIP driver to access remote PLCs connected in Ethernet/IP, DH+, RIO or ControlNet (Future Implementation) networks, via the 1756-ENET interface module



DRIVER STATION Field

When you configure the ABCIP driver to perform a routing to a remote network using a backplane with a 1756-ENET board, the syntax will be the following:

<Family>:<IP>:<BackPlane>:<Slot>:<Channel>:<Remote Node Address>

Where:

PLC Family:
1 – ControlLogix (Future)
2 or 500 – SLC500
3 or 5 – PLC5

Or

<Family>:<IP>

Where:

PLC Family:
4 or 1100 – Micrologix 1100

IP: IP address of the 1756-ENET Ethernet Card

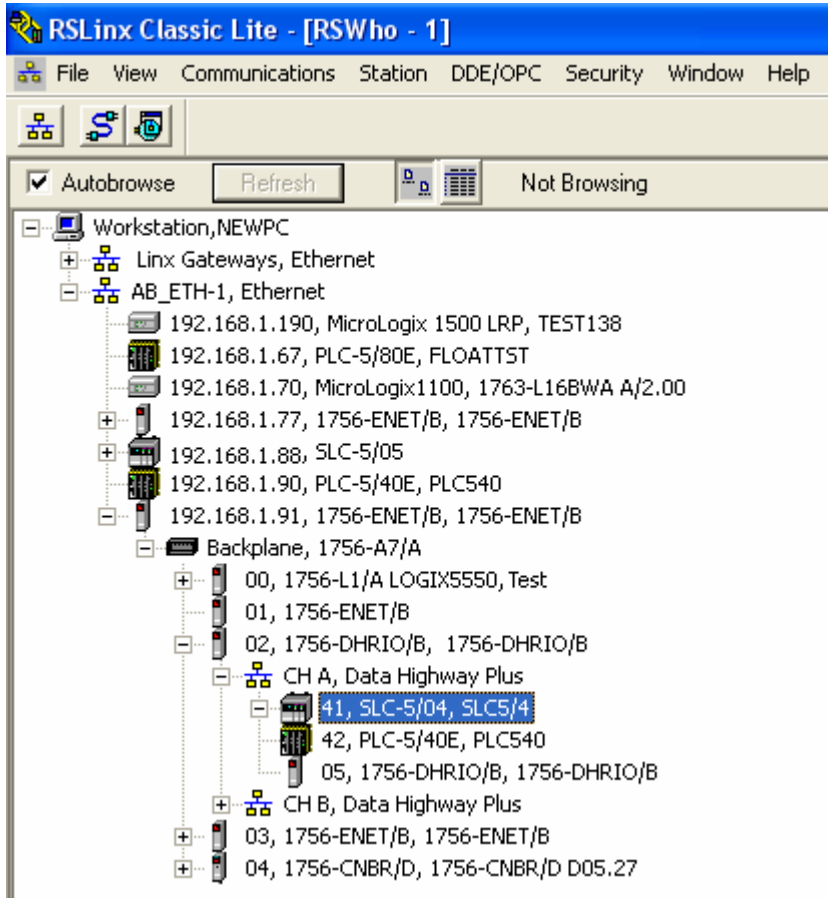
Backplane: Always 1

Slot: where the DH+ (1756-DHRIO) card (or CNet card) is

Channel: A or B – Channel where your DH+ Network is connected to that Remote Node Access

Remote Node Address: DH+ Address of the remote PLC in **Decimal**

For instance, in the picture below, we will access the highlighted PLC5/40 NSC_SYS1, DH+ address 41



The Station Address will be:

500: 192.168.1.91:1:2:A:33

Where:

500 – **SLC500 family**

192.168.1.91 – **Backplane** IP Address

1 - **Backplane** (always 1)

2 – **Slot** where the 1756-DHRIO/B is in the Backplane

A – **Channel** – CH A, Data Highway Plus

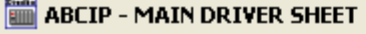
33 = 41 (octal) - **DH+ Address for the NSC_SYS1**

Driver Address field

When configuring the Addresses that the driver will be communicating with, you can follow the same syntax that is good for the drivers ABKE and ABCIP:

Header field: <File Type><File Number>:<Initial Address> (e.g N7:0, T4:0, and so forth)
Address field: <Offset to Initial Address> or
 <Offset to Initial Address>.<extension>, for Timers, Counters (extensions = PRE, ACC, CON, DN, EN, etc..) and Control (extensions = LEN, POS, CON)

Example:



Description:

Disable:

Read Completed: Read Status:

Write Completed: Write Status:

Min:
 Max:

| | Tag Name | Station | I/O Address | Action | Scan |
|---|-----------------|----------------------------|-------------|-----------------------------------------------------|-------------------------------------------------|
| 1 | SLC504_T4_0_ACC | 500: 192.168.1.91:1:2:A:33 | T4:0.ACC | Read+Write ▼ | Always ▼ |
| 2 | PLC5_B3_4_B1 | 5: 192.168.1.91:1:2:B:34 | B3:0/4 | Read+Write ▼ | Always ▼ |
| 3 | ML_N7_0 | 4:10.168.23.70 | N7:0 | Read+Write ▼ | Always ▼ |