



A-B RIO Adapter for SoftPLC® Runtime

Version 1.6 and 2.0

Table of Contents

1. Warranty	1
1.1. Terms of Use	1
2. Overview	2
2.1. Introduction	2
2.2. Concepts	2
2.3. Features	2
2.4. Required Hardware/Software	2
2.4.1. Hardware	2
2.4.2. Software	3
3. Usage	4
3.1. Driver Installation	4
3.2. Enable RIOSLAVE TLM	4
3.3. Configure TLM	5
3.3.1. RIOSLAVE.LST Configuration Editor Usage	5
4. Configuration	7
4.1. RIOSLAVE.LST Configuration File Details	7
4.1.1. RIOSLAVE (version 2.0, or later) Sample Configuration File	7
4.1.2. RIOSLAVE (legacy) Sample Configuration File	10
Appendix A: Hardware Overview	12
A.1. Smart-V2 SoftPLC Wiring	12
A.2. A-B RIO Wiring Connection to Din-Rail Terminal Block	13

Chapter 1. Warranty

1.1. Terms of Use

Because of the variety of uses of the information described in this manual, the users of, and those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the information. In no event will SoftPLC Corporation be responsible or liable for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

SOFTPLC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

SoftPLC Corporation reserves the right to change product specifications at any time without notice. No part of this document may be reproduced by any means, nor translated, nor transmitted to any magnetic medium without the written consent of SoftPLC Corporation.

SoftPLC, TOPDOC and Smartboard are registered trademarks of SoftPLC Corporation.

© Copyright 2011-2024 SoftPLC Corporation ALL RIGHTS RESERVED

First Printing May, 2011

Latest Printing September, 2025

SoftPLC Corporation
25603 Red Brangus Drive
Spicewood, Texas 78669

USA Telephone: 1-800-SoftPLC
WW Telephone: 512/264-8390
Fax: 512/264-8399
URL: <http://softplc.com>
Email: support@softplc.com

Chapter 2. Overview

2.1. Introduction

This document describes the installation, usage, and functionality of the **RIOSLAVE** TLM, which is a TOPDOC Loadable Module (TLM) that enables a SoftPLC to emulate remote racks on an Allen-Bradly Remote I/O network. The **RIOSLAVE** TLM implements the Allen-Bradley Remote I/O slave (adapter) protocol (RIO).

This functionality is only available on SoftPLC's equipped with an A-B RIO Adapter Interface. For older Smart SoftPLCs (firmware 4.x, or older) the part number for this interface is 'SM-ABRIOA#'. For newer Smart and NeoPAC SoftPLC/Gateways (firmware 5.x, or later) the part number for this interface is 'SPO-BH'. On all SoftPLC devices these interfaces are factory installed *options*.

2.2. Concepts

The **SoftPLC runtime engine** software supports TLM's, which are shared library extensions to SoftPLC. A TLM may be loaded either as a **DRIVER** or as a **MODULE**. The difference between a DRIVER and a MODULE is that a DRIVER is called once per SoftPLC scan, and optionally an additional number of times per scan. A MODULE is only called when the control program decides to call it and not as an inherent part of the scan. TLM's are made known to SoftPLC in the MODULES.LST file which may be edited by **TOPDOC NexGen** by traversing to: PLC | Modules.

This RIOSLAVE TLM is a DRIVER and has no TOPDOC Loadable Instructions.

2.3. Features

- Supports racks numbered from 0 to 076 (octal).
- Can emulate up to 63 racks concurrently, per port.
- Supports a 120 ohm termination resistor, software selectable.
- Supports baud rates of 57600 (56.7Kbaud), 115200 (115.2Kbaud), and 230400 (230.4Kbaud).
- Block transfer is supported.

2.4. Required Hardware/Software

2.4.1. Hardware

- Smart SoftPLCs (firmware 4.X), with
 - SM-ABRIOA5 and/or SM-ABRIOA6: Smart A-B RIO Slave Interface hardware (one per network).
 - SM-COM6CBLTB (required for SM-ABRIOA6 only) - Smart COM6 Port Cable to Din-Rail Terminal Block Assembly _Includes RJ11 interconnect cable and din-rail terminal block.
- Smart or NeoPAC (firmware 5.X) SoftPLCs/Gateways, with

- SPO-BH: A-B RIO Interface hardware (one per network).
- Belden #9463 Blue Hose connection cable(s) [customer supplied]



Only version 2.0 (or later) Smart boards can support more than one A-B RIO interface.

2.4.2. Software

- SoftPLC version 4.6 runtime, or later
- TOPDOC NexGen 1.6, or later

Chapter 3. Usage

To utilize the RIOSLAVE TLM, the following procedure shall first be completed. Steps 1-3 are described in detail within this chapter.

1. **TLM Installation** - Ensure the TLM is installed in the SoftPLC
2. **Enable TLM** – Configure SoftPLC to load the TLM at startup
3. **Configure TLM** – Edit the configuration file for your desired communications
4. **Configure other Device(s)** – Following vendor instructions, set up the other device(s) to communicate with the SoftPLC (not described in this document)

3.1. Driver Installation

The RIOSLAVE TLM's library file is named 'rioslave.tlm.so', and its configuration file is 'RIOSLAVE.LST'. The RIOSLAVE files should already be installed in the device's /SoftPLC/tlm directory. If they are not installed, please contact SoftPLC support. Please have the serial number of the specific device at the ready.

DRIVER UPDATES

SoftPLC drivers are occasionally updated for various reasons. In the vast majority of cases, these drivers may be updated in the field if the SoftPLC device is placed onto a network with internet access; if only temporarily, for the purposes of updating.

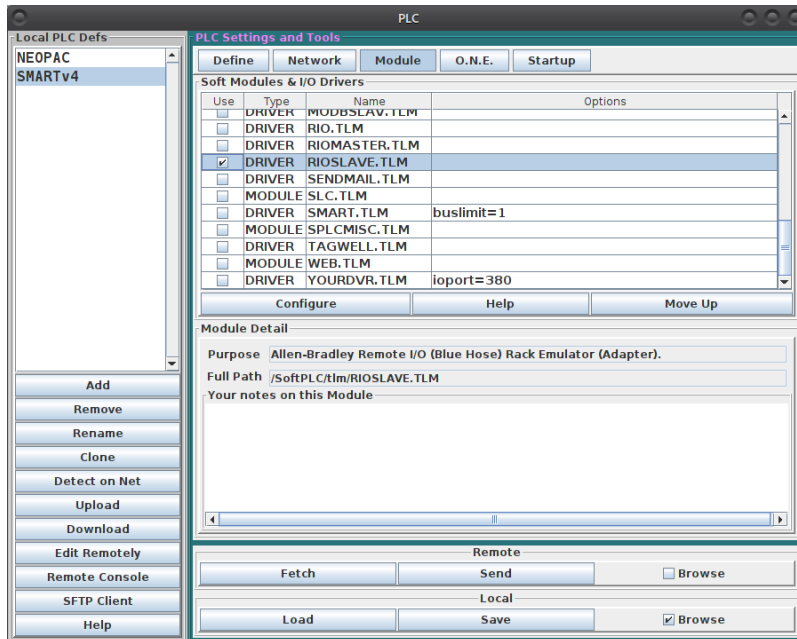
3.2. Enable RIOSLAVE TLM



The TOPDOC NexGen Manual and Help System describe using the Module Editor to enable and configure TLM's. The following sections assume previous knowledge of TOPDOC NexGen's Module Editor, and other SoftPLC configuration procedures.

- The [**Load**] button will load the file from the development system's disk.
- The [**Save**] button will write the file to the development system's disk.
- The [**Fetch**] button will load the file from the runtime system's disk.
- The [**Send**] button will write the file to the runtime system's disk.

To use the RIOSLAVE TLM, navigate to TOPDOC NexGen's **PLC | MODULE** editor. Then enable the TLM by filling the **Use** checkbox on the same row as the **RIOSLAVE.TLM** entry (**Save** and **Send** after filling the checkbox). Next, click on the **Configure** button.



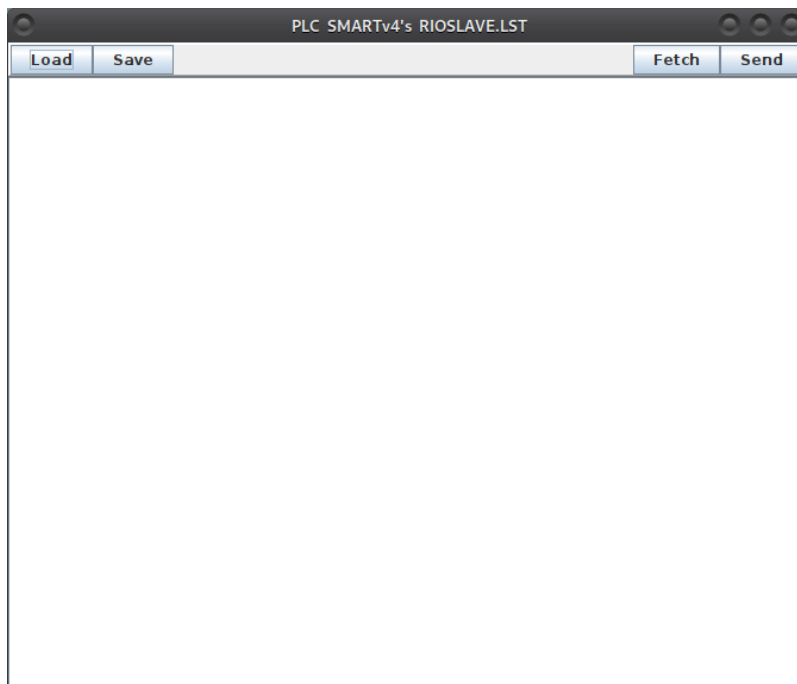
It is good practice to **both [Save] and [Send]** ANY edits, so that both the development system and the SoftPLC get a copy.

3.3. Configure TLM

The configuration file for the RIOSLAVE TLM is a text file called **RIOSLAVE.LST**.

3.3.1. RIOSLAVE.LST Configuration Editor Usage

With the RIOSLAVE.TLM line highlighted/selected, click the **[Configure]** button below the list of modules. This will load the Configuration Editor for the RIOSLAVE.LST file.



A template file is included in the SoftPLC. When connected to the SoftPLC, use the **[Fetch]** to load the template file into the Configuration Editor.

```
PLC SMARTv4's RIO SLAVE.LST
Load Save Fetch Send
# RIO SLAVE Driver Configuration for SoftPLC Runtime Software.
# This file uses an "s-expression" format, which is like XML but uses
# parentheses instead of angle bracketed keywords. The syntax of s-expression
# files is well documented and can be googled. The grammar, on the other hand,
# unlike the syntax, is application specific. SoftPLC uses '#' to indicate
# a single line comment. If the first non-blank character is #, then this
# line is a single line comment. Single line comments cannot follow
# any other token on the same line. Multiline 'C' style comments are also
# supported: /* followed by an eventual */
#
# S-expressions are thought to be more human readable than XML
# and are easy to parse with a programming language. Parentheses are balanced.
#
# The supported "grammar" is documented in the next several lines of comments.
#
# Grammar rules are declared like this, for example:
# (timeout_mult <4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 > )
# This means "timeout_mul" is a keyword and it can be followed with
# 4, 8, 16, 32, 64, 128, 256, or 512. The '|' means "or". e.g. (timeout_mult 4
#
# If something is wrapped in angle brackets <> in a grammar rule, this means
# it is a grouping. Only one of the choices in a grouping may actually be used.
# Angle brackets, being only part of the grammar definition, are never actually
# used in real configuration statements.
#
# If something is wrapped in square brackets [] in a grammar rule, this means
# whatever appears in the brackets is optional. If omitted, then a default value
# will be used and the default is defined in the grammar rule by a comment follow
ng
```

After RIO SLAVE.LST is properly edited given your application's specific configuration, execute a **Save** and **Send**. Finally, for the changes to take effect, there are three options:

1. Power cycle the SoftPLC.
2. Restart the SoftPLC via the PLC | Startup tab '**Restart SoftPLC Runtime...**' button.
3. You may enter "Remote Program" mode using the ladder online editor, then select "Remote Program" a second time. This pseudo transition from Remote Program to Remote Program is a signal to the TLM that it should reload its configuration file. This way you can reconfigure without cycling power, although it does require you enter "Remote Program" mode (twice!).

Chapter 4. Configuration

4.1. RIOSLAVE.LST Configuration File Details

The configuration file for the RIOSLAVE TLM is `/SoftPLC/tlm/RIOSLAVE.LST`. This file was changed upon the release of the driver's 2.0 version. Both versions are considered below.

In both versions, the file is used to set the debug level, baudrate, termination resistor on/off state, the I/O Bus Addresses and associated SoftPLC Datable assignments. Each of the configurable driver settings are explained by the *commented* lines shown in the configuration files themselves.

4.1.1. RIOSLAVE (version 2.0, or later) Sample Configuration File

```
# RIOSLAVE Driver Configuration for SoftPLC Runtime Software.
# This file uses an "s-expression" format, which is like XML but uses
# parentheses instead of angle bracketed keywords. The syntax of s-expression
# files is well documented and can be googled. The grammar, on the other hand,
# unlike the syntax, is application specific. SoftPLC uses '#' to indicate
# a single line comment. If the first non-blank character is #, then this
# line is a single line comment. Single line comments cannot follow
# any other token on the same line. Multiline 'C' style comments are also
# supported: /* followed by an eventual */
#
# S-expressions are thought to be more human readable than XML
# and are easy to parse with a programming language. Parentheses are balanced.
#
# The supported "grammar" is documented in the next several lines of comments.
#
# Grammar rules are declared like this, for example:
# (timeout_mult <4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 > )
# This means "timeout_mul" is a keyword and it can be followed with
# 4, 8, 16, 32, 64, 128, 256, or 512. The '|' means "or". e.g. (timeout_mult 4)
#
# If something is wrapped in angle brackets <> in a grammar rule, this means
# it is a grouping. Only one of the choices in a grouping may actually be used.
# Angle brackets, being only part of the grammar definition, are never actually
# used in real configuration statements.
#
# If something is wrapped in square brackets [] in a grammar rule, this means
# whatever appears in the brackets is optional. If omitted, then a default value
# will be used and the default is defined in the grammar rule by a comment following
# a ':'. The square brackets, being only part of the grammar definition,
# are never actually used in real configuration statements.
#
# White spaces including tabs and newlines (CR, LF, etc.) are all ignored.

#----- Grammar Rules: -----
# (rioslave                                ` the master containing element.
#                                           ` every nested element is contained within
```

```

#           ` this master element.
# (driver_version VERSION_NUM) ` VERSION_NUM must be 2
# (port PORT_NUMBER ...) ` see below for (port).
# [(port PORT_NUMBER ...) ...] ` (port) may occur multiple times.
# ) ` end of (rioslave)
#
# Details on nested elements:

# Element (port ...)
# contains data for a particular bluehose port. It will consist of one or more
# (rack_part) elements.
#
# (port PORT_NUMBER           ` PORT_NUMBER says which bluehose port to use
#                             ` Smart4: 0, 1, or 2.
#                             ` NeoPAC: 1 or 2
# (baudrate BAUDRATE)       ` BAUDRATE must be 57600, 115200 or 230400
# [(termination_resistor <yes|no>)] ` default: "no"
# [(debug BIT_MAP)]         ` BIT_MAP may be 0 or non-zero and
#                             ` controls debug logging statements
#                             ` default: 0
# (rack_part ...)           ` See below for (rack_part)
# [(rack_part ...) ...]     ` (rack_part) may occur multiple times
# ) ` end of (port)

# Element (rack_part)
# contains data for a particular adapter. A (rack_part) normally corresponds to
# an adapter and will take an Allen-Bradley remote IO address range length of
# either 2, 4, 6 or 8 words in an octal numeric format. The data transfered to
# an adapter is mapped into the SoftPLC datatable for both directions: inputs
# and outputs. This is done with four different types of elements, two for
# digital data and two for block transfer data: (inputs), (outputs), (bt_read),
# and (bt_write).
#
# (rack_part <OCTAL_RACK_NUM><STARTING_WORD_IN_RACK>-<OCTAL_RACK_NUM><LAST_WORD_IN_RACK>
# (inputs ...) ` see below, optional and at most one
# (outputs ...) ` see below, only one allowed
# [(bt_read ...) ...] ` see below, multiple allowed
# [(bt_write ...) ...] ` see below, multiple allowed
# ) ` end of (rack_part)

# Element (inputs)
# contains the SoftPLC datatable address where the adapter emulation should retrieve
# input words from. The range (count) of words at the supplied starting DT_ADDRESS is
# implicit in the (rack_part) word range. For example, (rack_part 010-017 ...)
# says that 8 words are mapped, numbered 0 thru 7. Or (rack_part 126-127 ...)
# says 2 words are mapped, again starting at DT_ADDRESS given in the (inputs) element.
# (inputs DT_ADDRESS
# ) ` end of (inputs)

# Element (outputs)
# contains the SoftPLC datatable address where the adapter emulation should stuff
# output words to. The range (count) of words at the supplied starting DT_ADDRESS is

```

```

# implicit in the (rack_part) word range.  For example, (rack_part 010-017 ...)
# says that 8 words are mapped, numbered 0 thru 7.  Or (rack_part 126-127 ...)
# says 2 words are mapped, again starting at DT_ADDRESS given in the (outputs) element.
# (outputs DT_ADDRESS
# )                ` end of (outputs)

# Element (bt_read)
# contains a mapping of an I/O slot in the main processor's memory to a datatable
# memory block in this slave machine.
# (bt_read <OCTAL_RACK_NUM><WORD_IN_RACK>,<0|1>
#                ` WORD_IN_RACK range: 0 to 7.
#                ` e.g. 063,1 and in this example:
#                ` OCTAL_RACK_NUM is 06,
#                ` group is WORD_IN_RACK is 3, and
#                ` slot comes after the comma and is 1
#                ` indicating the hi-byte (1 for hi byte/slot,
#                ` 0 for low byte/slot)
# (src DT_ADDRESS) ` starting datatable word address of block
# (size WORD_COUNT) ` WORD_COUNT is no. words to xfer in the read
#                ` range: 1 to 64
# )                ` end of (bt_read)

# Element (bt_write)
# contains a mapping of an I/O slot in the main processor's memory to a datatable
# memory block in this slave machine.
# (bt_write <OCTAL_RACK_NUM><WORD_IN_RACK>,<0|1>
#                ` see (bt_read) above
# (dst DT_ADDRESS) ` starting datatable word address of block
# (size WORD_COUNT) ` WORD_COUNT is no. words to xfer in the write
#                ` range: 1 to 64
#                ` If the master tries to write more than this
#                ` then the excess words are not written.
# )                ` end of (bt_write)

# end of intro comments and grammar rules.
# Next comes actual configuration statements:

(rioslave (driver_version 2)
  (port 1
    (baudrate 57600)
    #(debug 0x1)
    #(termination_resistor yes)

    (rack_part 020-027 (inputs N17:0)(outputs N7:0)
      (bt_write 022,0 (dst N7:300) (size 10) )
      (bt_read 022,0 (src N17:008)(size 64) )
      (bt_write 022,1 (dst N7:310) (size 10) )
      (bt_read 022,1 (src N17:100)(size 64) )
    ) /* end of rack_part */

    (rack_part 044-045 (inputs N17:8)(outputs N7:8))

```

```

)          /* end of port */

(port 2
  (rack_part 010-011 (inputs N117:0)(outputs N100:0))
)
)          /* end of rioslave */

```

4.1.2. RIOSLAVE (legacy) Sample Configuration File

```

; Configuration file for SoftPLC TLM "RIOSLAVE".
; Comments start with a semicolon.

[DRIVER]
; Set to 0 normally, 1 for full RIO network logging (using console redirection)
DEBUG=2

; Which COM port, either 5 or 6 are typically allowed for a Smart.
; And 0 is allowed for a Neo
INTERFACE=0

; BAUDRATE may be one of: 57600, 115200 or 230400
BAUDRATE=57600

; TERMINATION_RESISTOR may be yes or no, and possibly adds a 120 ohm
; termination resistor via this software setting
TERMINATION_RESISTOR=no

; For PLC-2 master only:
; if this TLM is acting as a last adapter, set to yes
; else set to no
LAST_ADAPTER=no

; A block of 1 word(s) that hold(s):
; [0]: RIO Master Mode
STATUS_WORDS=N17:200

; IOBusAddr Defined:
;
; IOBusAddr is used in the rows below, and is rack number and an even io group
; address of the start of a quarter rack of AB RIO. For example:
;
; I:036
; ^ ==^
; | ^ |
; | | +----- I/O group number, must be even, one of: 0,2,4, or 6
; | |
; | +----- 2 digit octal rack number 0 to 76, but further limited by
; | your RIO master, i.e. by your PLC. We start from rack zero.

```

```

; |
; +----- I or 0 meaning read or write to the rack, respectively.
;           Use I under [INPUTS], and 0 under [OUTPUTS]

; The IOBusAddr are the addresses actually used on the RIO cabling systems, and
; are therefore relative to the master RIO scanner.

```

```

[INPUTS]
;DatatableAddr      IOBusAddr      NumWords
I:004                I:024                4

```

```

[OUTPUTS]
;DatatableAddr      IOBusAddr      NumWords
O:004                O:024                4

```

```

; [BLOCKTRANSFER_READS] and [BLOCKTRANSFER_WRITES] sections create a mapping of
; an I/O slot in the main processor's memory to a datatable memory block in this
; slave machine. From 1 to 64 words can be transfered for each BT READ or WRITE,
; and for both slot=0 and slot=1 of each I/O word.

```

```

[BLOCKTRANSFER_READS]
;DatatableAddr      IOBusAddr      NumWords (1-64)      Slot (0-1)
;N7:0100            I:027                64                    1

```

```

[BLOCKTRANSFER_WRITES]
;N10:0              O:024                10                    1

```

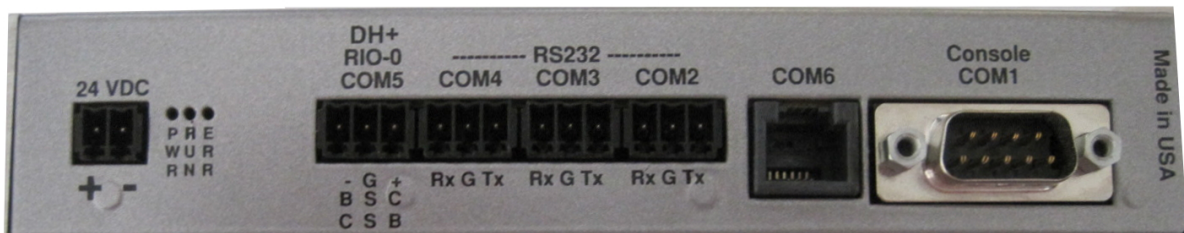
Appendix A: Hardware Overview

Optional RIO interface

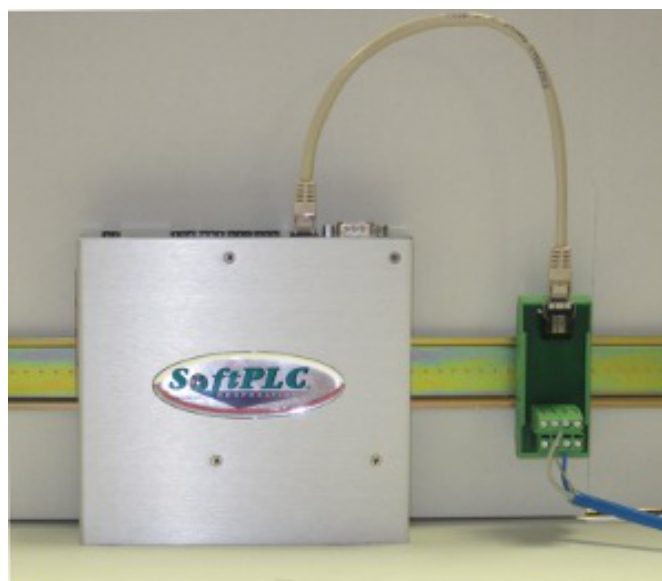
The A-B RIO interface is optional, internal hardware; requiring factory installation into the SoftPLC device. For the older firmware devices (4.X, and older), the **SM-ABRIOA5** replaces the COM5 port and the **SM-ABRIOA6** is available for COM6. For newer firmware devices (5.X, and newer), the **SPO-BH** is available. Without this hardware, successful use of the RIOSLAVE driver is not possible. Call SoftPLC Support to discuss possible options for adding this hardware to an existing device.

A.1. Smart-V2 SoftPLC Wiring

The figure below shows the port/wiring connection for the blue hose connected to COM5 of a Smart (firmware 4.X, or older) SoftPLC with SM-ABRIOM5. Using the center line of text, the Clear wire connects to the terminal marked “C”, the Shield connects to the center marked “S”, and the Blue wire connects to the terminal marked “B”.

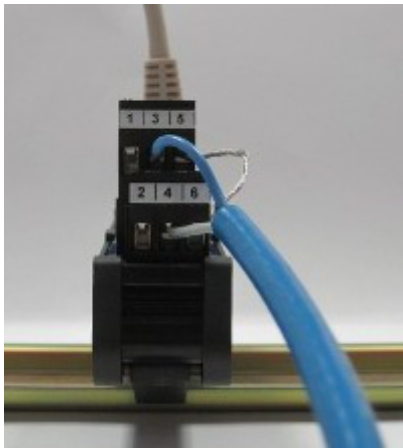


The image below shows a mounted Smart SoftPLC attached to the Smart COM6 Cable Assembly (Cat No SM-COM6CBLTB). Smart (version 1.x and 2.x) boards support RIO via COM6.



A.2. A-B RIO Wiring Connection to Din-Rail Terminal Block

The Terminals labeled by number in the image below correspond to the following wiring:



1. No Connection
2. No Connection
3. Blue
4. Clear
5. Shield/Drain
6. No Connection