SCADAPack E Target 5 I/O
Device Reference

Documentation

Table of Contents

I SCADAPack E Target 5 I/O Device Reference

Schneider Electric

Documentation

Printed in Canada.

Version: 8.05.4

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

1 Technical Support

Support related to any part of this documentation can be directed to one of the following support centers.

Technical Support: The Americas

Available Monday to Friday 8:00am – 6:30pm Eastern Time

Toll free within North America	1-888-226-6876
Direct Worldwide	+1-613-591-1943
Email	TechnicalSupport@controlmicrosystems.com

Technical Support: Europe

Available Monday to Friday 8:30am – 5:30pm Central European Time

Direct Worldwide	+31 (71) 597-1655
Email	euro-support@controlmicrosystems.com

Technical Support: Asia

Available Monday to Friday 8:00am – 6:30pm Eastern Time (North America)

Direct Worldwide	+1-613-591-1943
Email	TechnicalSupport@controlmicrosystems.com

Technical Support: Australia

Inside Australia	1300 369 233
Email	au.help@schneider-electric.com

2 Safety Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

	The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.
	This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, **can result** in death or serious injury.

⚠CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result** in minor or moderate injury.

CAUTION

CAUTION used without the safety alert symbol, indicates a potentially hazardous situation which, if not avoided, **can result in** equipment damage..

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and the installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

SCADAPack Workbench and SCADAPack E Smart RTU are not suitable for controlling safety-critical systems. SCADAPack Workbench and SCADAPack E Smart RTU are not tested for, nor have approval for use in, the control of safety-critical systems. Safety-critical systems should be controlled by an approved safety-critical platform that is independent of SCADAPack Workbench and SCADAPack E Smart RTU.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Do not control safety-critical systems with SCADAPack Workbench and SCADAPack E Smart RTU.

Failure to follow these instructions can result in death, serious injury or equipment damage.

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠CAUTION
EQUIPMENT OPERATION HAZARD
• Verify that all installation and set up procedures have been completed. • Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices. • Remove tools, meters, and debris from equipment.
Failure to follow these instructions can result in injury or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and grounds, except those grounds installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove ground from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or

unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.

- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

3 Preface

Scope

This manual describes in detail each I/O Device configuration provided with the SCADAPack Workbench.

The SCADAPack E Smart RTU provides on-device hardware I/O channels as well as RTU database points which can be connected to IEC 61131-3 resources. Resources can also be used to interface with other third party RTUs such as the Allen Bradley and Modbus PLCs. As such, the I/O devices provided with the allow access to onboard or remote I/O data.

Purpose

The purpose of this document is to describe the custom I/O devices provided with the SCADAPack Workbench.

Detailed information for the I/O device drivers for the Allen Bradley and Modbus devices is available in the *SCADAPack E Target 5 DF1 PLC Interface*, and *SCADAPack E Target 5 Modbus Communication Interfaces* manuals.

Assumed Knowledge

Familiarity with the SCADAPack Workbench is strongly recommended.

Target Audience

- Systems Engineers
- Commissioning Engineers
- Maintenance Technicians

References

- *SCADAPack E Configuration Reference*
- *SCADAPack E Target 5 Modbus Communication Interfaces*
- *SCADAPack E Target 5 DF1 PLC Interface*
- SCADAPack E Technical Reference Manuals.
- Workbench Help

4 Overview

This manual describes in detail each I/O Device configuration provided with the SCADAPack Workbench.

This manual is to be used along with the *SCADAPack E Target 5 Technical Reference*. Summary information is provided in this manual for communication using the I/O interfaces to basic SCADAPack E RTU I/O, as well as Modbus & DF1 protocol devices.

For additional information on how to configure Modbus & DF1 devices, consult the respective SCADAPack E PLC Communication manuals.

In addition to the SCADAPack E RTU I/O complex equipment interfaces, the I/O device library provides three major types of I/O connection to the SCADAPack E RTU data.

- I/O devices with the ***RTU*** prefix, presented in Section ***RTU I/O Devices*** 15 - are used to access groups of SCADAPack E RTU points data such as physical I/O or derived points.
- Function blocks for RTU data access, presented in Section ***RTU Data via Function Blocks*** 11 - are used to access data points programmatically rather than via I/O devices.
- I/O devices with the ***MBUS, MTCP*** or **MRTUTCP** prefix presented in sections ***Serial Modbus Master I/O Devices*** 47, ***Modbus TCP Client I/O Devices*** 66 ***Modbus RTU in TCP Client I/O Devices*** 83 and are used to access to Modbus data on a peripheral PLC devices connected to the SCADAPack E RTU via a serial or TCP connection. This section of the manual may be used in conjunction with the *SCADAPack E Target 5 Modbus Communication Interfaces* manual.
- I/O devices with other prefixes represent I/O devices for PLC or peripheral devices. For example:

  DF1 prefix presented in Section ***Allen Bradley PLC I/O Devices*** 99 are used to access to data on the Allen- Bradley family of PLC's. This section of the manual can be used in conjunction with the *SCADAPack E Target 5 DF1 PLC Interface* manual. AB PLC's supported include:

  - SLC 500 Series
  - PLC 5 Series
  - DF1 Generic PLC's

5 Variable / RTU Point Interaction

Variables attached to the I/O devices that are presented throughout this manual read data from or write data to the SCADAPack E RTU database points. *READ* devices read data from the RTU database into input variables. *WRITE* devices write data to the RTU database from output variables.

Each I/O device has a parameter corresponding to the first variable on the device (called "First_Point_Number" or "First_Register"). This may be a point number, or PLC register of the first point of the device. The following configuration concepts and rules apply:

- Variables attached to I/O devices correspond to consecutively numbered items. If reading from the RTU point database, the I/O address may be any valid RTU data point corresponding to physical, derived or RTU system data of a compatible type. If reading data from a peripheral Modbus type PLC, the I/O address may be any valid Modbus register corresponding to physical I/O.
- BOOL or BIN Boolean I/O devices correspond to consecutive binary points starting at the address of the I/O device. Boolean Output devices cannot reference "read-only" RTU data points (e.g. physical digital inputs). Boolean I/O devices support multiple channels.
- DINT or REAL Analog I/O devices support multiple channels; each corresponds to an RTU analog

value. The I/O address represents the address of the first data item (e.g. RTU point number, PLC register address) depending on the device type. Subsequent channels correspond to consecutive items. Analog Output devices cannot reference "read-only" data registers (e.g. physical analog inputs).

- READ_OUTPUT Input devices support multiple channels; each corresponds to the status of a RTU physical output point.
- COUNTER_READ devices map to RTU counter input points or system counter points. RTU counters are managed internally by 32-bit unsigned data types and are presented in 32-bit format to analog integers.
- The RTU String Output device writes an STRING variable type to a system string point.

6 RTU Data via Function Blocks

The SCADAPack E Smart RTU provides a mechanism separate from I/O devices (described in the following sections) for accessing RTU data.

In general the function blocks require more processing capacity in the RTU compared with I/O devices, but provide greater programming flexibility and access to more detailed RTU data.

Function blocks provide access to reading and writing current data values to/from the RTU point database, as well as access to point attributes not available via I/O devices.

- The **GETPNTxx** functions and function blocks allow an application to read point current value data from the database (applies to Physical I/O, Derived data and System Points)
- The **SETPNTxx** functions allow an application to write point current value data to the database (applies to Physical Outputs, Derived data and System Points)
- The **RTUCROB** function blocks allow an application to have accurate pulse control of binary points (applies to Physical Binary Outputs and Derived Binary points)
- The **RDFLD_x** function blocks allow an application to read attribute and property fields from points in the RTU database
- The **SETATR_x** function blocks allow an application to set attributes of points in the RTU database
- The **RDREC_x** function blocks provide applications with a set of commonly used attribute and property fields for points in the RTU database

Arrays can be useful when using the above Functions and Function Blocks. For more information see *SCADAPack E Target 5 Technical Reference* manual.

Details of these, and other SCADAPack E Smart RTU function blocks are described in the *SCADAPack E Target 5 Function Block Reference* manual.

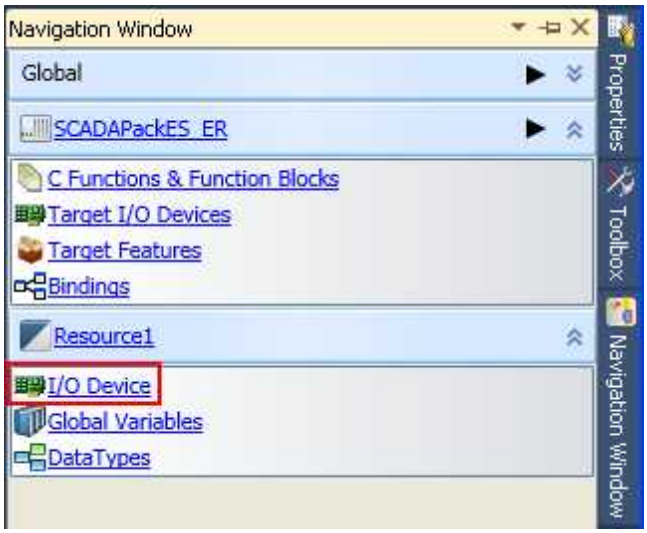
7 I/O Devices

Physical inputs and outputs on the SCADAPack E RTU can be accessed by the application via the I/O Device connections.

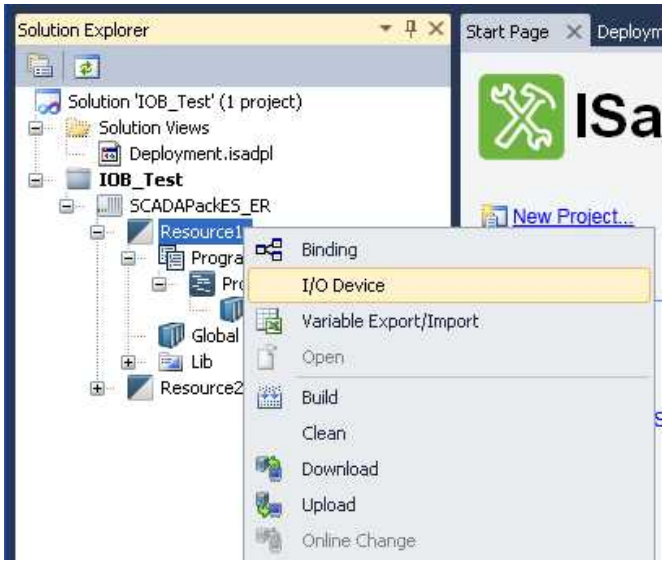
SCADAPack E RTU internal data points may be accessed via I/O devices (or via other C Function Blocks - e.g. **GETPNTxx** and **SETPNTxx** functions and function blocks).

Each I/O device needs to be supplied with an address that specifies the RTU starting point number when reading from inputs or writing to outputs. This address is entered into the *first_point_num* or *first register* field of the particular I/O device within the I/O Wiring tool.

I/O devices are accessible within a project by clicking on the *I/O Device* menu from the Workbench *Navigation Window*.



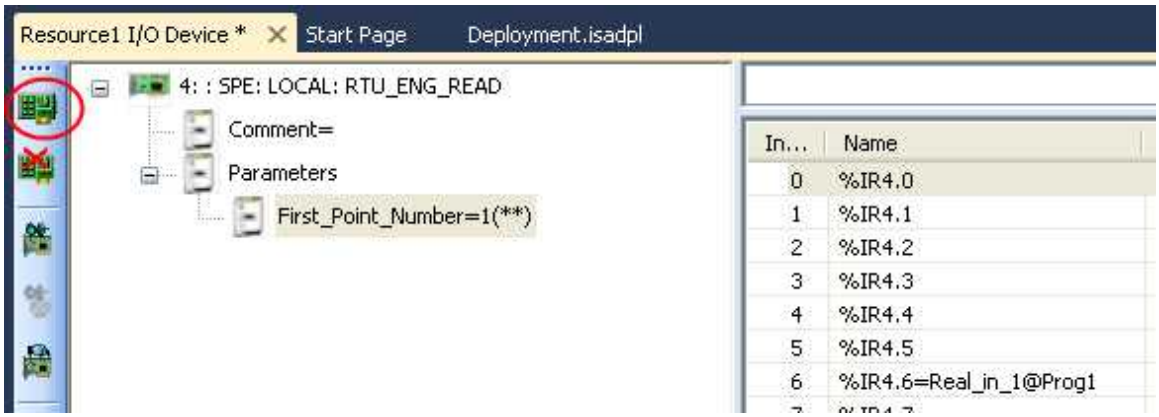
Or by right-clicking on a *Resource* item in the Workbench *Solution Explorer* window.



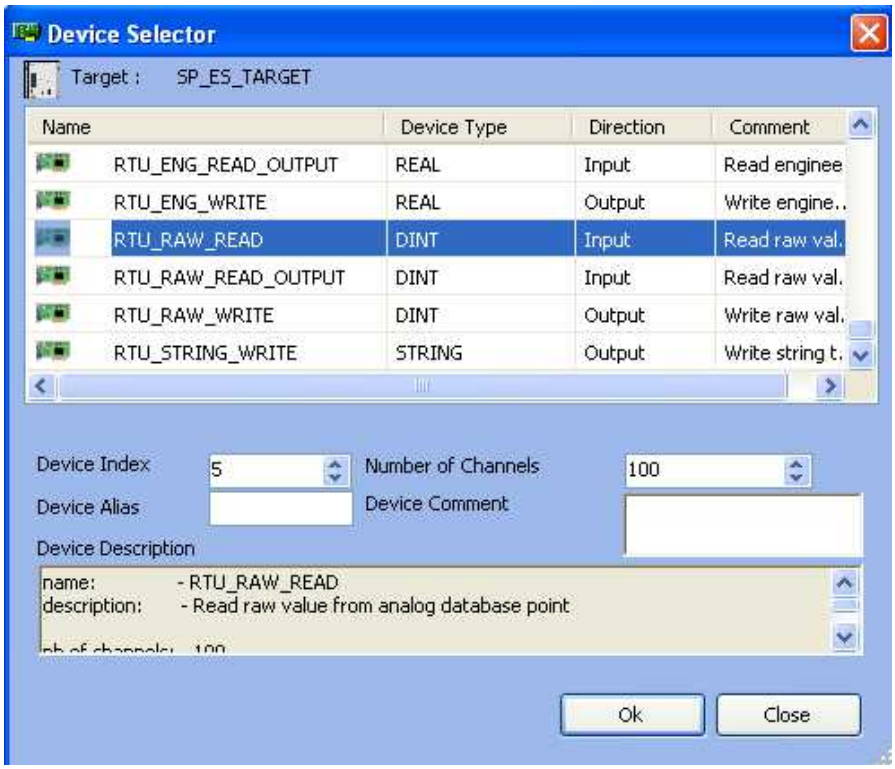
Either of these should display the ***I/O Device Window*** 14.

7.1 I/O Device Window

The *I/O Device Window* allows the user the add and edit I/O Devices that are associated with the selected Resource.



Selecting the *Add Device* icon from the I/O Wiring toolbar (shown above) displays the *I/O Device Selector* dialog.



I/O devices contain multiple channels having the same type and direction. When adding I/O devices, the *Device Selector* enables selecting from those available for the target. You specify a device index and a number of channels. You can also include an alias name and comment. The device index value can range from 0 and 65535. Devices are shown in the *I/O Device Window* by their Device Index order.

The default Number of Channels in the I/O Device settings represents the maximum permitted for the

device type. Adjust this to the desired number of channels for this device.

7.2 RTU Database I/O Devices

This section presents the I/O devices used to access database points configured in the SCADAPack E Smart RTU. Both physical I/O points and derived points may be accessed from an I/O device.

RTU I/O Devices provide a convenient and efficient way to interface with groups of points in the RTU's point database. Alternative mechanisms for interfacing with RTU point data includes the use of **GETPNTxx** and **SETPNTxx** function blocks. These function blocks have the advantage of accessing point data programatically but are less efficient, accessing one point at a time.

I/O devices need not necessarily correspond to the RTU I/O card arrangements.

These I/O devices are listed in the *I/O Device Selector* dialog with the **RTU** prefix (e.g. RTU_RAW_READ).

7.2.1 I/O Devices for RTU Binary Input / Digital Input Points

Physical RTU digital inputs have one interface. The state of physical digital input points is read through RTU_BIN_READ devices. The state of a physical digital input point cannot be controlled.

Derived binary input points are read through RTU_BIN_READ devices. Derived Binary input points are controlled through RTU_BIN_WRITE_INPUT I/O devices.

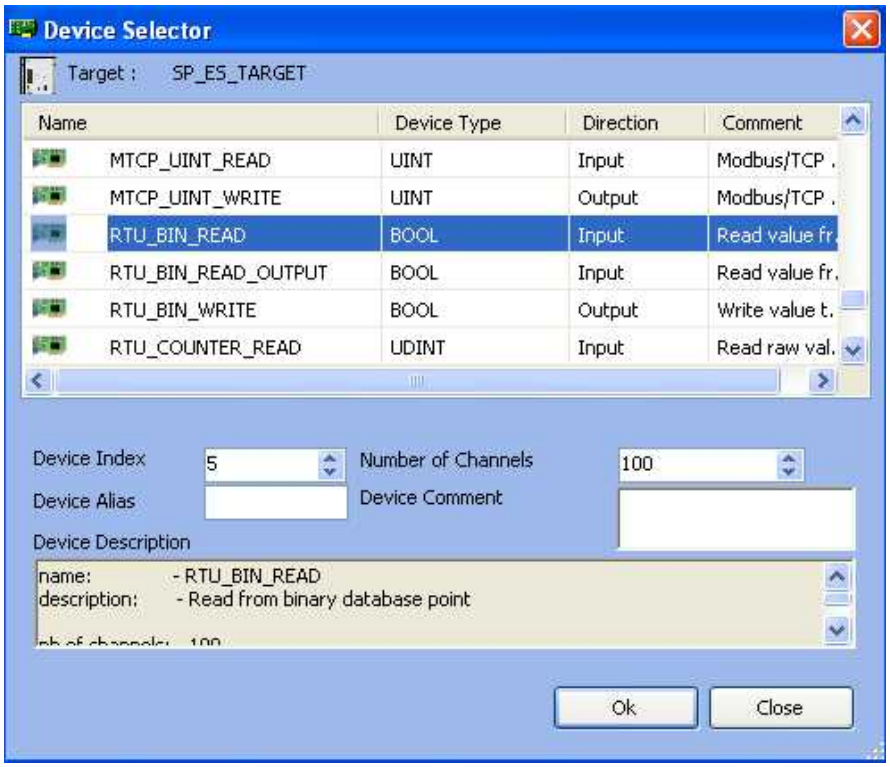
Where an application attaches a Boolean variable to a RTU_BIN_WRITE_INPUT device, the *Current State* property of the digital point will be controlled from the attached variable.

7.2.1.1 Reading RTU Binary Input / Digital Input Points

RTU_BIN_READ Devices

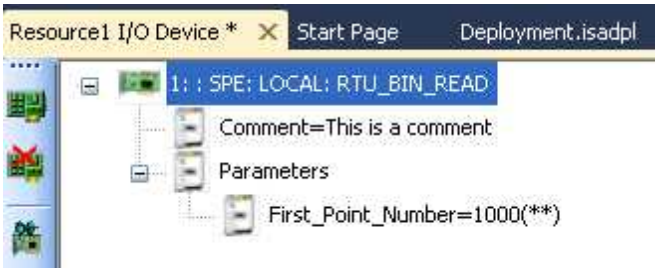
RTU Physical digital input points and derived binary input points may be read through the *RTU_BIN_READ* I/O Device and read data from the RTU point database.

Where an application attaches a BOOL *VarInput* type variable to an RTU_BIN_READ I/O Device, the *Current State* property of the digital point will be read into the variable. If the digital point is a Physical Binary point address, the physical digital input channel corresponding to that address is read.



The *RTU_BIN_READ* I/O Device has a user-selectable Number of Channels between 1 and 100.

Associated with each **RTU** device is a *FIRST_POINT_NUMBER* field that assigns the SCADAPack E database physical binary input or derived binary input point number to the first Channel on the device. Variables assigned to subsequent channels are assigned to consecutive point numbers.

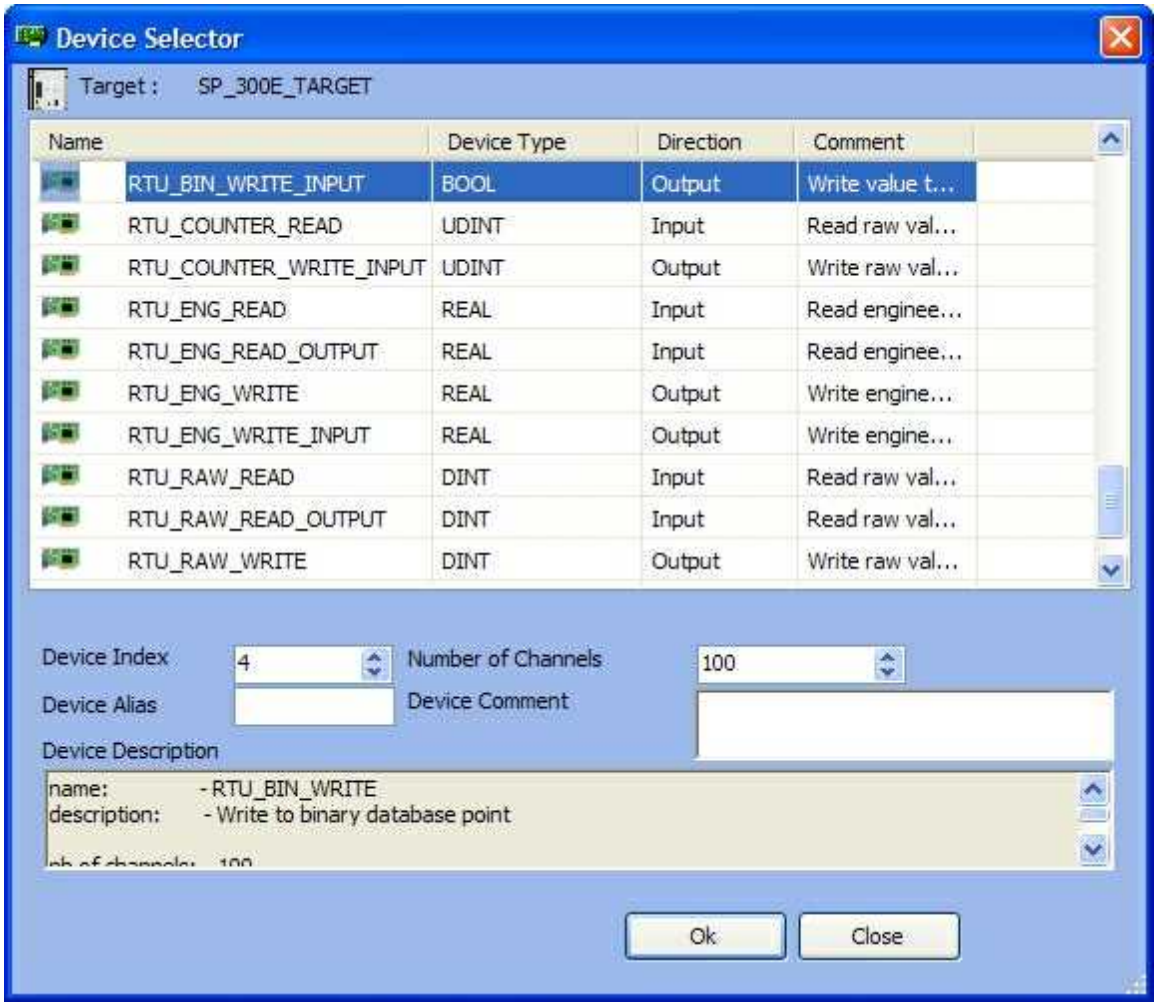


RTU point objects supported include Physical Inputs, Derived & System Binary Objects. Connected variables are updated at the start of the scan with the *Current State* property of the digital point from the RTU point database.

7.2.1.2 Writing RTU Binary Input / Digital Input Points

RTU_BIN_WRITE_INPUT Devices

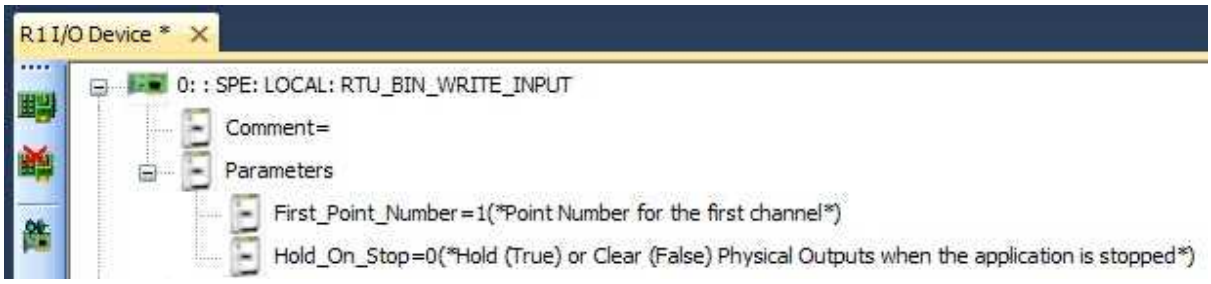
The Channels on this device type are connected to BOOL *VarOutput* variables within an application and write data to points in the RTU point database.



The *RTU_BIN_WRITE_INPUT* I/O Device has a user-selectable Number of Channels between 1 and 100.

Associated with each device are the following parameters:

- ***First_Point_Number*** field assigns the SCADAPack E database derived binary point number to the first channel on the device. Subsequent channels are assigned to consecutive database point numbers of the same type
- ***Hold_On_Stop*** field affects RTU point database Physical Output point types only. This parameter has no effect on this I/O device.



RTU point objects supported for output from these devices include Derived Binary points & System Binary points.

The *Current State* property of the binary input points (in the RTU point database) is updated with the state of the variables at the end of the scan.

RTU database points that correspond to channels on an output device that are not connected to variables are typically set to the OFF state by programs. Even though a point is not connected to a variable on the I/O device, the points do behave as if they are under application control. If it is desired to control the unconnected points external to a resource (e.g. via DNP3, Modbus, IEC60870, etc) reduce the Number of Channels parameter on the device to avoid overlap.

7.2.2 I/O Devices for RTU Binary Output / Digital Output Points

Physical RTU digital outputs have two sets of interfaces. The state of a digital output is controlled through the RTU_BIN_WRITE I/O devices. The feedback status of a physical digital output is read through RTU_BIN_READ_OUTPUT devices.

Derived RTU Binary points are controlled through RTU_BIN_WRITE I/O devices. The feedback status of derived binary points is read into RTU_BIN_READ I/O devices.

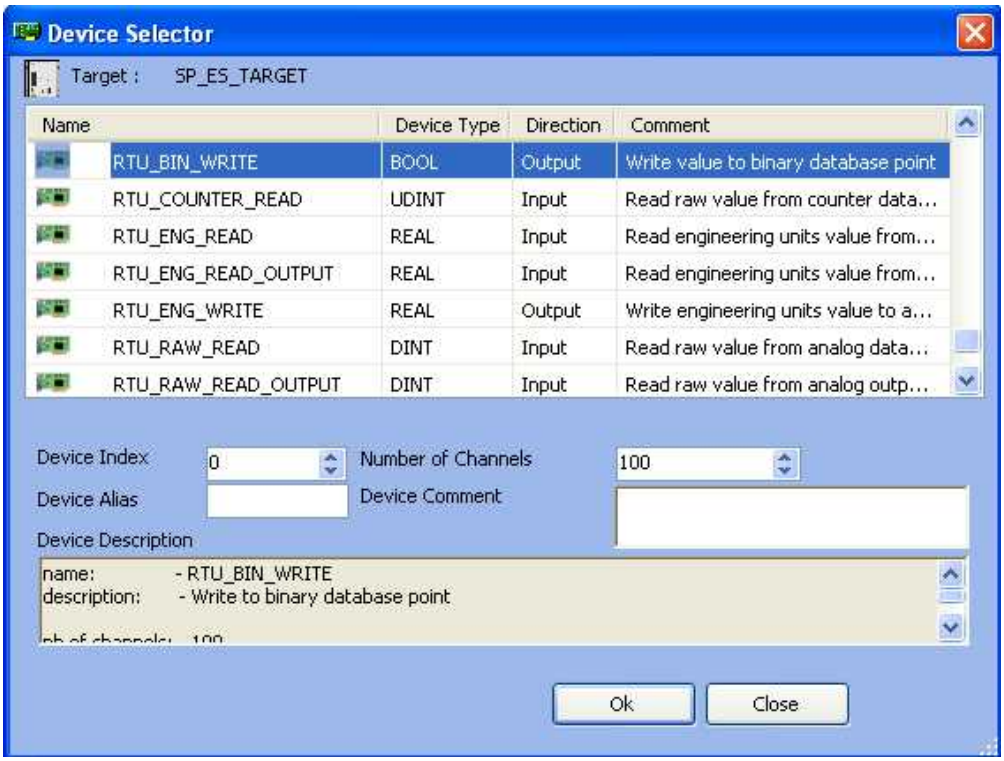
Where an application attaches a Boolean variable to a RTU_BIN_WRITE device, the *Current State* property of the digital point will be controlled from the attached variable.

- ***Writing to RTU Binary Output Points*** 19
- ***Reading from RTU Binary Output Points*** 21

7.2.2.1 Writing RTU Binary Output Points

RTU_BIN_WRITE Devices

The Channels on this device type are connected to BOOL *VarOutput* variables within an application and write data to points in the RTU point database.



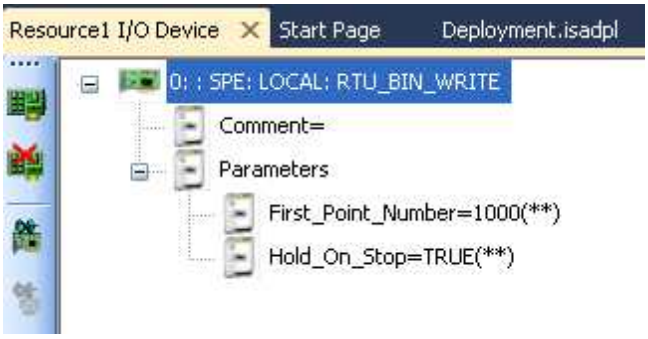
The *RTU_BIN_WRITE* I/O Device has a user-selectable Number of Channels between 1 and 100.

Associated with each device are the following parameters:

- ***First_Point_Number*** field assigns the SCADAPack E database physical output or derived binary point number to the first channel on the device. Subsequent channels are assigned to consecutive

database point numbers of the same type

- ***Hold_On_Stop*** field affects RTU point database Physical Output point types only. Setting this field to *true* holds physical output points defined on this device in their current state when the application is stopped. Setting this field to *false* (which is the default) resets the physical output points on this device to the OFF state when the application is stopped



RTU point objects supported for output from these devices include Physical Binary Output points, Derived Binary points & System Binary points.

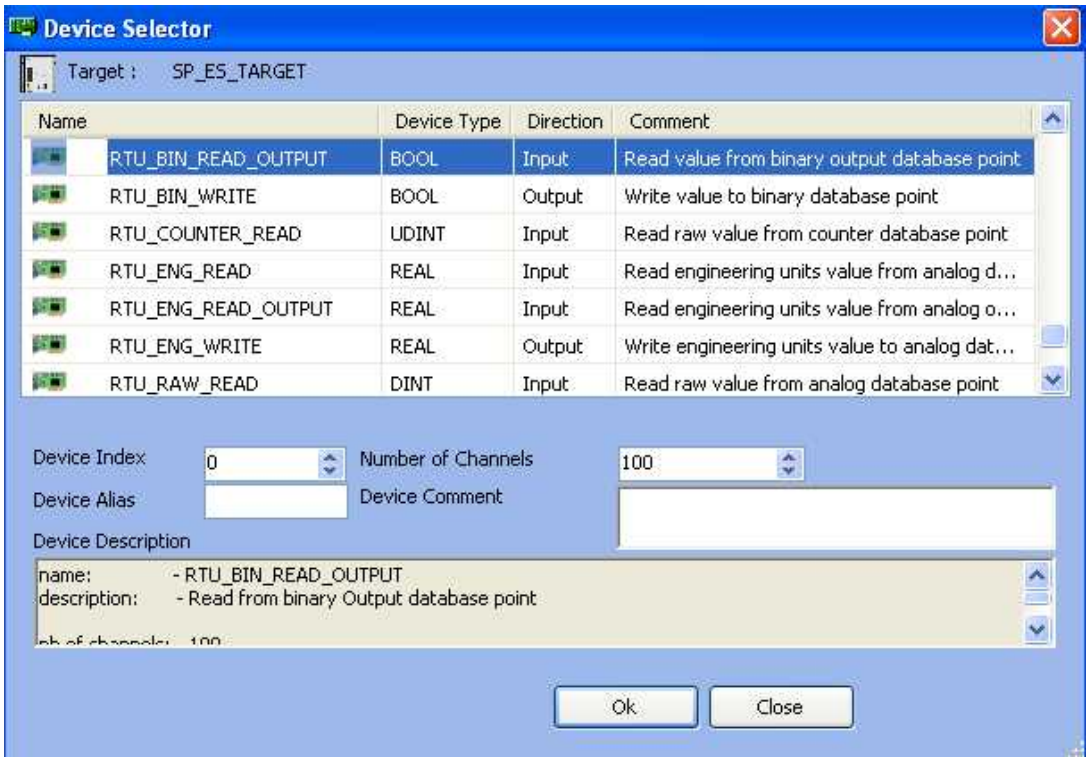
The *Current State* property of the binary output points (in the RTU point database) is updated with the state of the variables at the end of the scan.

RTU database points that correspond to channels on an output device that are not connected to variables are typically set to the OFF state by programs. Even though a point is not connected to a variable on the I/O device, the points do behave as if they are under application control. If it is desired to control the unconnected points external to a resource (e.g. via DNP3, Modbus, IEC60870, etc) reduce the Number of Channels parameter on the device to avoid overlap.

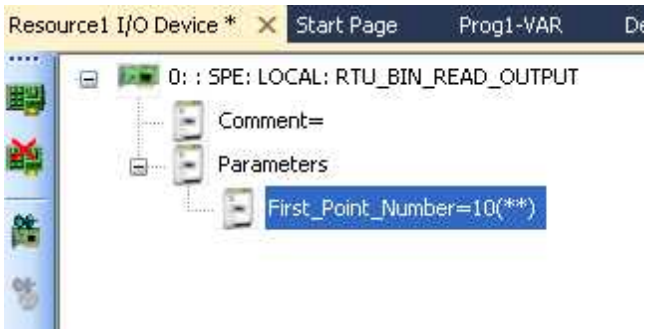
7.2.2.2 Reading RTU Binary Output Points

RTU_BIN_READ_OUTPUT Devices

This is an input device for reading output point state information back into an application. The Channels on this device are connected to BOOL *VarInput* variables within an application and provide feedback on the output state of RTU database binary output points.



Associated with each device is a ***First_Point_Number*** field that assigns a SCADAPack E database Binary Output point number to the first channel on the device. Variables assigned to subsequent channels are assigned to consecutive point numbers.



RTU point objects supported include Physical Outputs, Derived & System Binary Output Objects. Connected variables are updated at the start of the scan with the *Current State* property of the output point from the RTU point database.

There is a one to one mapping between the channels on the digital output device and feedback from digital output points. In order words, the first channel on the *RTU_BIN_READ_OUPUT* device will return the fed back *Current State* property of the first database point referenced by the device, and so on for each channel on the device. As a result, the digital output feedback devices pass input data based on the status of the digital output back into an application. This feature can be used to track a discrepancy between an application output and the current state of the actual digital output being controlled by the variable.

7.2.3 Analog I/O Devices / Point Representation & Conversion

Analog RTU Read & Write I/O Device types are available to have either Integer (IEC DINT) or floating point (IEC REAL) variables attached. Both DINT and REAL analog variables are represented in 32-bit format. The RTU data interface to these devices is accomplished via point properties in the RTU point database. In addition to direct variable data mapping, conversion tables may be attached to any analog Input/Output variable. Conversion table functions are applied after the following conversion rules are applied:

- An DINT variable attached to an RTU_RAW_READ or RTU_RAW_READ_OUTPUT device Channel receives a 32-bit signed value from the point's *Current Integer Value* property. The type of DNP3 object selected for this point does **not** affect the value presented to the application (i.e. An analog point's value may have a conversion applied to a 16-bit DNP3 analog object, but the conversion is not applied to the value reported to resources).
- An REAL analog (floating point) variable attached to an RTU_ENG_READ or RTU_ENG_READ_OUTPUT device Channel receives a 32-bit floating point value from the point's *Current Engineering Value* property. The type of DNP3 object selected for this point does not affect the value presented to resources
- An DINT variable attached to an RTU_RAW_WRITE device Channel sends a 32-bit signed value to the point's *Current Integer Value* property. A conversion between integer and engineering value is also carried out according to an integer to engineering conversion formula. The type of DNP3 object selected for this point does not affect the value presented from resources
- An REAL analog (floating point) variable attached to an RTU_ENG_WRITE device Channel sends a 32-bit floating point value to the point's *Current Engineering Value* property. A conversion between engineering and 32-bit floating point value is also carried out according to engineering to integer conversion formula. The type of DNP3 object selected for this point does not affect the value presented from resources.
- An UDINT variable attached to a Counter Input device receives an unsigned 32-bit integer value representing the count value of an RTU counter point.

Due to the arrangement of RTU data mapping for physical I/O, *VarInput* variables attached to physical I/O points on RTU_BIN_READ, RTU_RAW_READ or RTU_ENG_READ I/O devices read the state or value of the physical Input points. *VarOutput* variables attached to physical I/O points on RTU_BIN_WRITE, RTU_RAW_WRITE or RTU_ENG_WRITE I/O devices control or write to the physical output points. To read the status of physical output points, attach *VarInput* variables to RTU_BIN_READ_OUTPUT, RTU_RAW_READ_OUTPUT or RTU_ENG_READ_OUTPUT I/O devices.

INT, DINT, UINT and REAL analog variables may **not** be mixed on the same Analog I/O Device. The appropriate conversion functions may be used as necessary.

7.2.4 I/O Devices for RTU Analog Input Points

Physical RTU Analog Inputs have one set of interfaces. The state of an analog output is read through RTU_RAW_READ or RTU_ENG_READ I/O devices. The state of a physical analog input point cannot be controlled.

The state of derived analog points is read through RTU_RAW_READ or RTU_ENG_READ I/O devices. Derived RTU Analog input points are controlled through RTU_RAW_WRITE_INPUT or RTU_ENG_WRITE_INPUT I/O devices.

Where an application attaches a DINT *VarOutput* variable to an RTU_RAW_WRITE_INPUT device, the *Current Integer Value* property of the analog point will be controlled from the variable. The analog point's *Current Integer Value* property, RAW-MIN, RAW-MAX, ENG-MIN & ENG-MAX attributes will be used to automatically calculate the *Current Eng.Value* property of the point.

Where an application attaches a REAL (floating point) *VarOutput* variable to an RTU_ENG_WRITE_INPUT I/O device, the *Current Eng. Value* property of the analog point will be controlled from the variable. The analog point's *Current Eng. Value* property RAW-MIN, RAW-MAX, ENG-MIN & ENG-MAX attributes will be used to automatically calculate the *Current Integer Value* property of the analog point.

I/O Devices are *strongly-typed*. Both DINT and REAL analog variables may **not** be mixed on the same Analog I/O device. However, conversion functions such as ANY_TO_DINT or ANY_TO_REAL may be used to perform a data type conversion in the user application as necessary.

- Reading RTU Analog Input Points 24
- Writing RTU Analog Input Points 26

7.2.4.1 Reading RTU Analog Input Points

RTU Analog points may be imported through RTU_RAW_READ or RTU_ENG_READ I/O devices. Where an application attaches a DINT variable to a Channel on a RTU_RAW_READ device, the *Current Integer Value* property of the analog point will be read into the variable. Where an application attaches a REAL (floating point) analog variable to an RTU_ENG_READ device Channel, the *Current Eng.Value* property of the analog point will be read into the variable. Where the analog point is a Physical Analog point address, the Physical Analog Input channel corresponding to that address is read. See **I/O Devices for RTU Analog Output Points** 28 regarding reading the value of a Physical Analog Output channels.

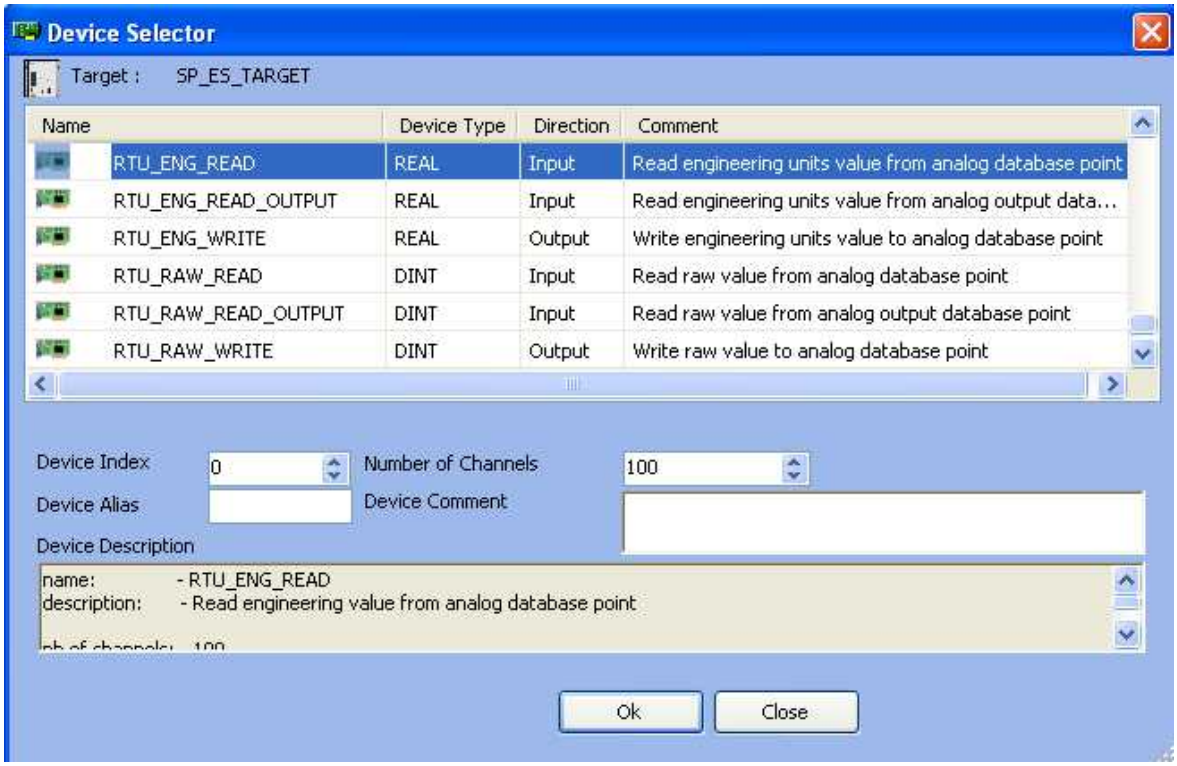
I/O Devices are *strongly-typed*. Both DINT and REAL analog variables may **not** be mixed on the same Analog I/O device. However, conversion functions such as ANY_TO_DINT or ANY_TO_REAL may be used to perform a data type conversion in the user application as necessary.

DINT variables contain signed 32-bit numbers. The value of an "Integer" analog variable will be the physical analog input variable in the range RAW-MIN to RAW-MAX as configured in the point's attributes.

REAL variables contain 32-bit floating point numbers. For a physical analog input variable, variables will be in the range ENG-MIN to ENG-MAX as configured in the point's attributes.

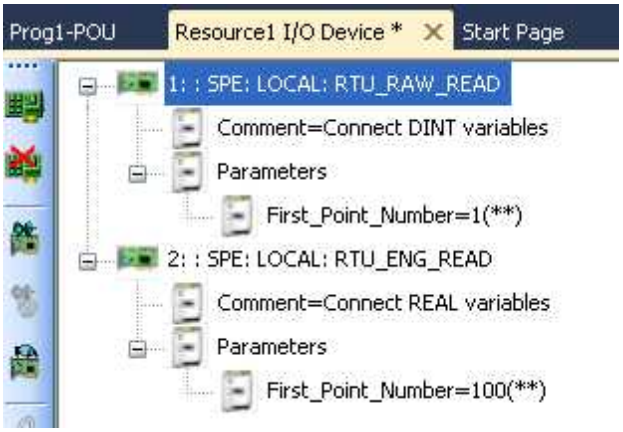
RTU_RAW_READ & RTU_ENG_READ Devices

Analog input devices are provided in the SCADAPack E I/O *Device Selector* as follows.



The RTU_RAW_READ and RTU_ENG_READ I/O devices have a user-selectable Number of Channels between 1 and 100.

Associated with each device is a ***First_Point_Number*** field that assigns the SCADAPack E database physical analog input or derived analog input point number to the first channel on the device. Variables assigned to subsequent channels are assigned to consecutive point numbers.



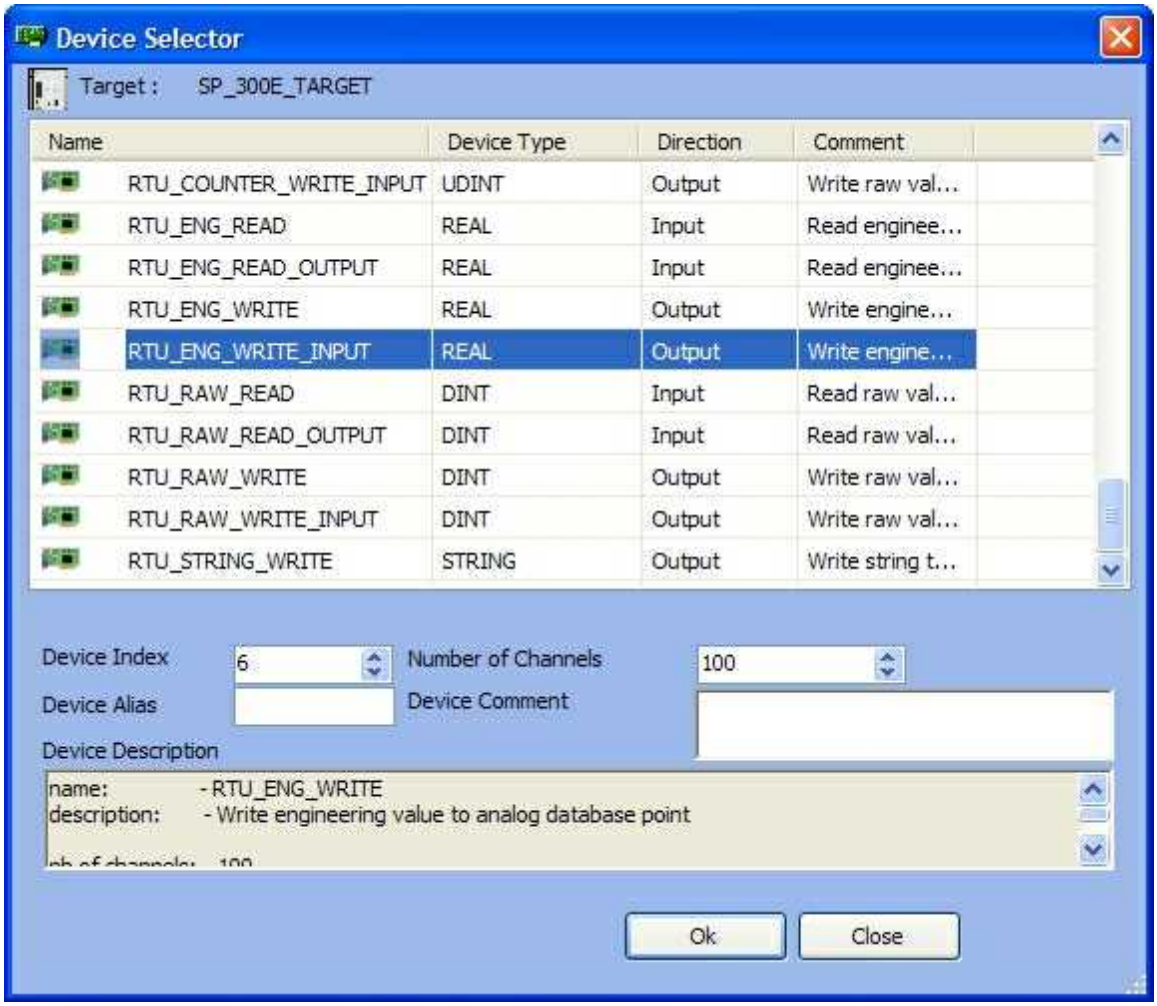
RTU point objects supported include Physical inputs, Derived Integer and Floating Point objects. Variables of type DINT are continuously updated with the *Current Integer Value* property whereas

variables of type REAL are updated with the *Current Eng Value* property from the RTU point database.

7.2.4.2 Writing RTU Analog Input Points

RTU_RAW_WRITE_INPUT and RTU_ENG_WRITE_INPUT Devices

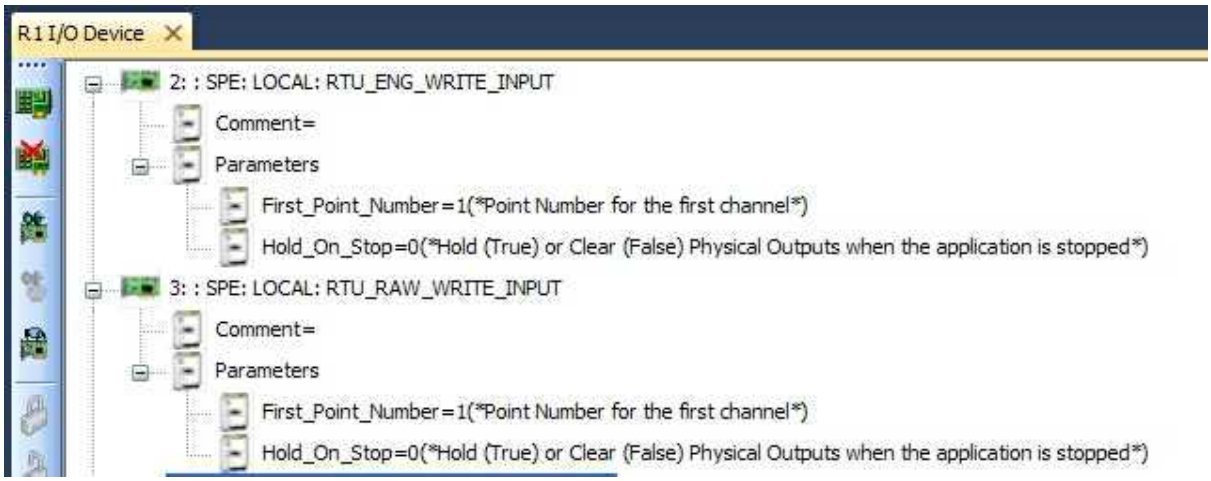
The Channels on these device types are connected to DINT or REAL *VarOutput* variables within an application and write data to points in the RTU point database.



The *RTU_RAW_WRITE_INPUT* and *RTU_ENG_WRITE_INPUT* I/O Devices have a user-selectable Number of Channels between 1 and 100.

Associated with each device are the following parameters:

- ***First_Point_Number*** field assigns the SCADAPack E database derived analog point number to the first channel on the device. Subsequent channels are assigned to consecutive database point numbers of the same type
- ***Hold_On_Stop*** field affects RTU point database Physical Output point types only. This parameter has no effect on this I/O device.



RTU point objects supported for output from these devices include Derived Analog points & System Analog points.

The *Current State* property of the analog input points (in the RTU point database) is updated with the state of the variables at the end of the scan.

RTU database points that correspond to channels on an output device that are not connected to variables are typically set to the OFF state by programs. Even though a point is not connected to a variable on the I/O device, the points do behave as if they are under application control. If it is desired to control the unconnected points external to a resource (e.g. via DNP3, Modbus, IEC60870, etc) reduce the Number of Channels parameter on the device to avoid overlap.

7.2.5 I/O Devices for RTU Analog Output Points

Physical RTU Analog Outputs have two sets of interfaces. The value of physical analog outputs is controlled through RTU_RAW_WRITE or RTU_ENG_WRITE I/O devices. The feedback status of an analog output is read through RTU_RAW_READ_OUTPUT or RTU_ENG_READ_OUTPUT I/O devices.

Derived RTU Analog points are controlled through RTU_RAW_WRITE or RTU_ENG_WRITE I/O devices. The feedback status of derived analog points is read through RTU_RAW_READ or RTU_ENG_READ I/O devices.

Where an application attaches a DINT *VarOutput* variable to an RTU_RAW_WRITE device, the *Current Integer Value* property of the analog point will be controlled from the variable. The analog point's *Current Integer Value* property, RAW-MIN, RAW-MAX, ENG-MIN & ENG-MAX attributes will be used to automatically calculate the *Current Eng.Value* property of the point.

Where an application attaches a REAL (floating point) *VarOutput* variable to an RTU_ENG_WRITE I/O device, the *Current Eng. Value* property of the analog point will be controlled from the variable. The analog point's *Current Eng. Value* property RAW-MIN, RAW-MAX, ENG-MIN & ENG-MAX attributes will be used to automatically calculate the *Current Integer Value* property of the analog point.

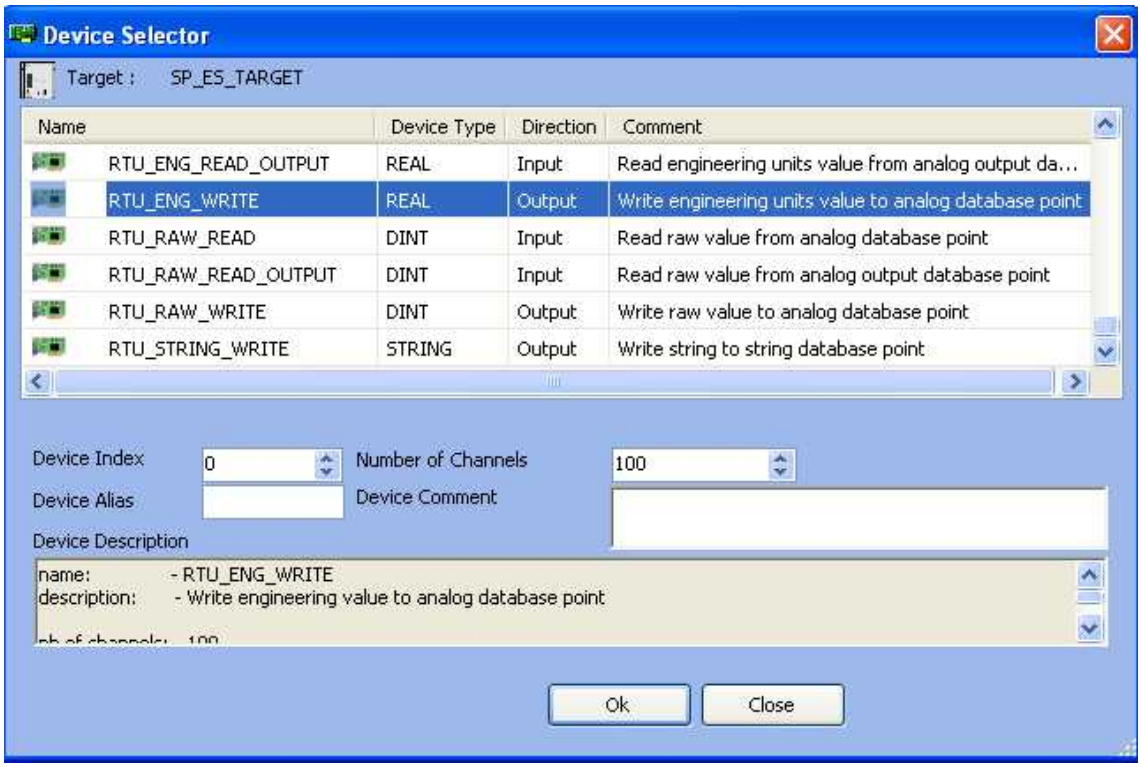
I/O Devices are *strongly-typed*. Both DINT and REAL analog variables may **not** be mixed on the same Analog I/O device. However, conversion functions such as ANY_TO_DINT or ANY_TO_REAL may be used to perform a data type conversion in the user application as necessary.

- ***Writing Analog Output Points*** 29
- ***Reading RTU Analog Output Points*** 31

7.2.5.1 Writing Analog Output Points

RTU_RAW_WRITE & RTU_ENG_WRITE Devices

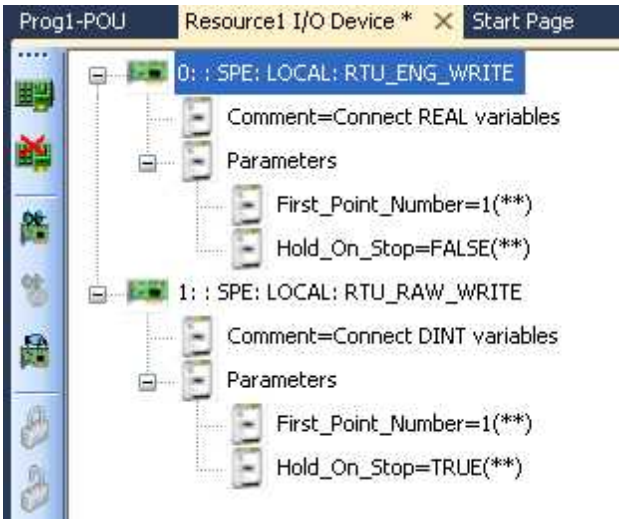
The Channels on these device types are connected to DINT and REAL *VarOutput* variables within an application and write data to points in the RTU point database.



The RTU_RAW_WRITE and RTU_ENG_WRITE I/O devices have a user-selectable Number of Channels between 1 and 100.

Associated with each device are the following parameters:

- ***First_Point_Number*** field assigns the SCADAPack E database physical output or derived analog point number to the first channel on the device. Subsequent channels are assigned to consecutive database point numbers of the same type
- ***Hold_On_Stop*** field affects RTU point database Physical Output point types only. Setting this field to *true* holds physical output points defined on this device in their current value when the application is stopped. Setting this field to *false* (which is the default) resets the physical output points on this device to a value of zero when the application is stopped



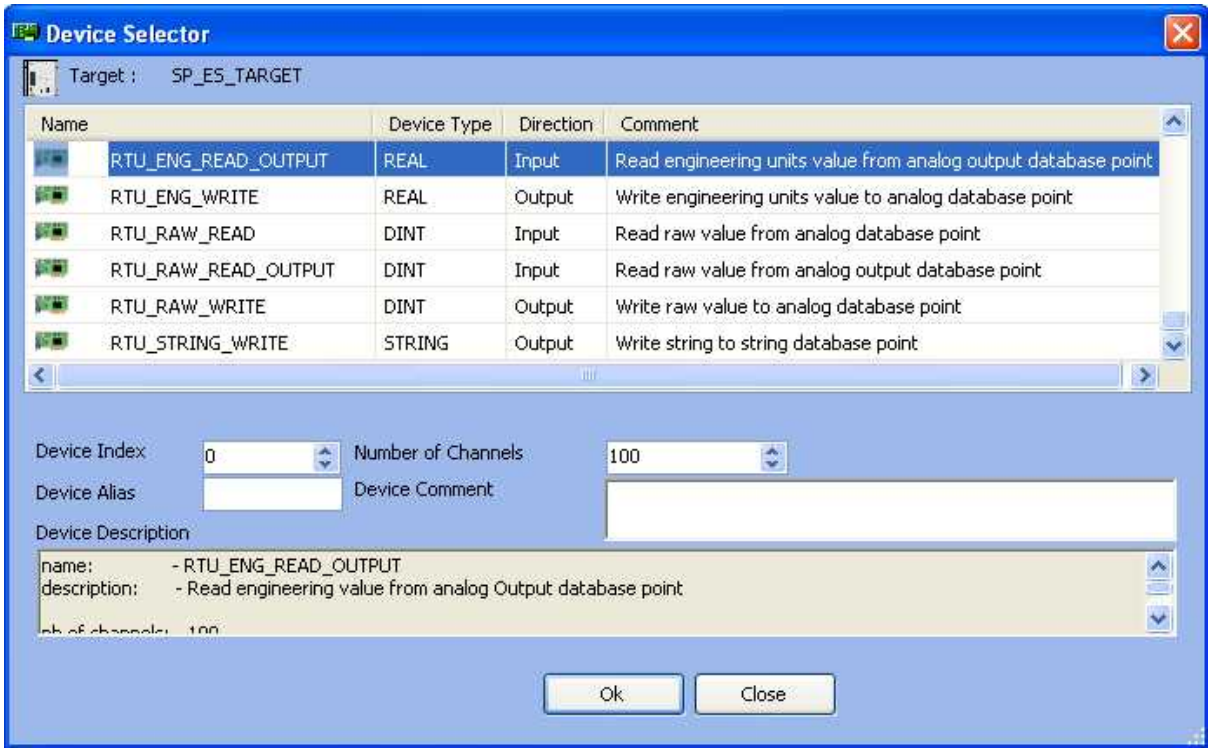
RTU points supported include Physical outputs, Derived Integer and Floating Point objects. At the end of the scan, variables of type DINT update the *Current Integer Value* property of the database point, whereas variables of type REAL updated the *Current Eng Value* property from the RTU point database.

INT, DINT, UINT and REAL analog variables may **not** be mixed on the same Analog I/O Device. The appropriate conversion functions may be used as necessary.

7.2.5.2 Reading RTU Analog Output Points

RTU_RAW_READ_OUTPUT & RTU_ENG_READ_OUTPUT Devices

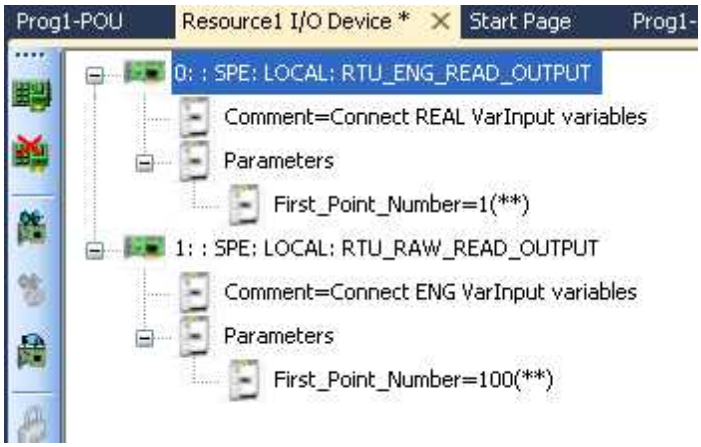
These are Input devices for retrieving RTU analog output point value information back into an application.



The RTU_RAW_READ_OUTPUT and RTU_ENG_READ_OUTPUT I/O devices have a user-selectable Number of Channels between 1 and 100.

These devices are connected to DINT or REAL Input variables within an application and provide feedback on the output state of RTU database analog output points.

Associated with each device is a ***First_Point_Number*** field that assigns a SCADAPack E database Analog Output point number to the first channel on the device. Variables assigned to subsequent channels are assigned to consecutive point numbers.



Connected variables are updated at the start of the scan with a *Current Integer Value* or *Current Eng Value* property of the output point from the RTU point database.

There is a one to one mapping between the channels on the I/O device and feedback from analog output points. In other words, the first channel on the analog I/O device will return the fed back *Current Integer* or *Eng Value* property of the first database point referenced by the device, and so on for each channel on the device. As a result, the analog output feedback devices pass input data based on the status of the analog output back into an application. This feature can be used to track a discrepancy between an application output and the current value of the actual analog output being controlled by the variable.

INT, DINT, UINT and REAL analog variables may **not** be mixed on the same Analog I/O Device. The appropriate conversion functions may be used as necessary.

7.2.6 I/O Devices for RTU Counter Input Points

Physical RTU counter inputs have one interface. The state of physical counter input points is read through RTU_COUNTER_READ devices. The state of a physical counter input point cannot be controlled.

Derived counter input points are read through RTU_COUNTER_READ devices. Derived Binary counter points are controlled through RTU_COUNTER_WRITE_INPUT I/O devices.

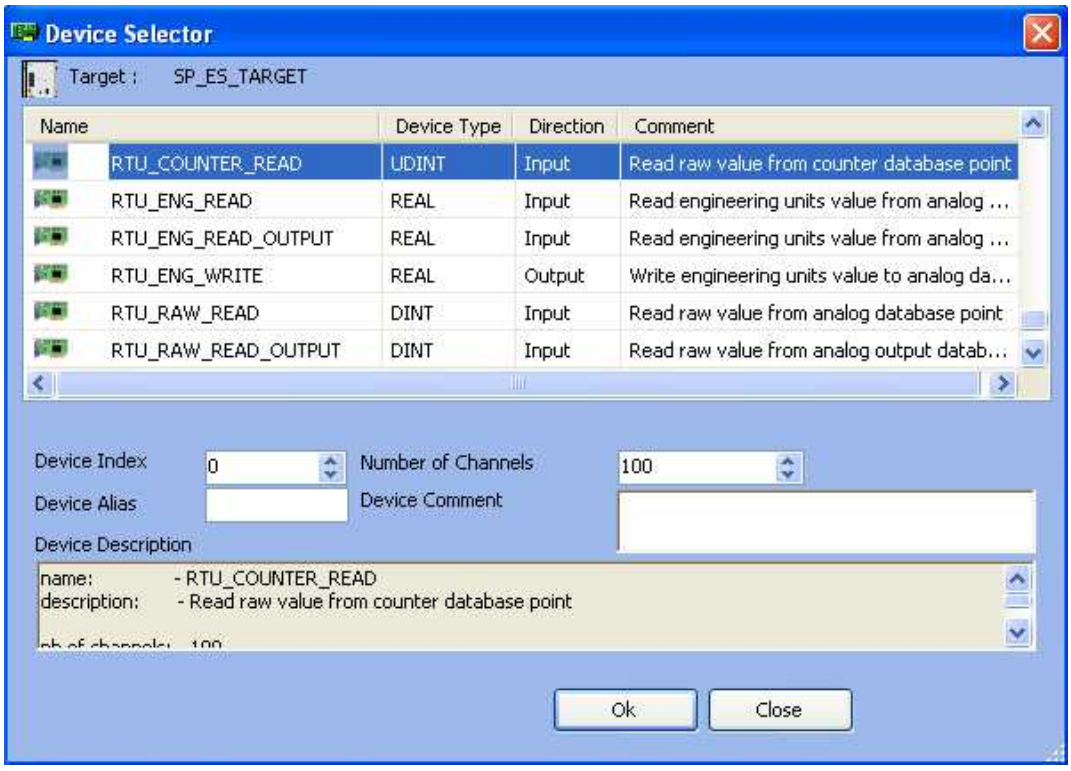
Where an application attaches a variable to a RTU_COUNTER_WRITE_INPUT device, the *Current State* property of the counter point will be controlled from the attached variable.

- Reading RTU Counter Input Points 33
- Writing RTU Counter Input Points 35

7.2.6.1 Reading RTU Counter Input Points

RTU_COUNTER_READ Devices

RTU_COUNTER_READ I/O Devices support only UDINT 32-bit Unsigned analog variables. The *Current Integer Value* property of the RTU physical counter input will be read into the variable.



The RTU_COUNTER_READ I/O Device has a user-selectable Number of Channels between 1 and 100.

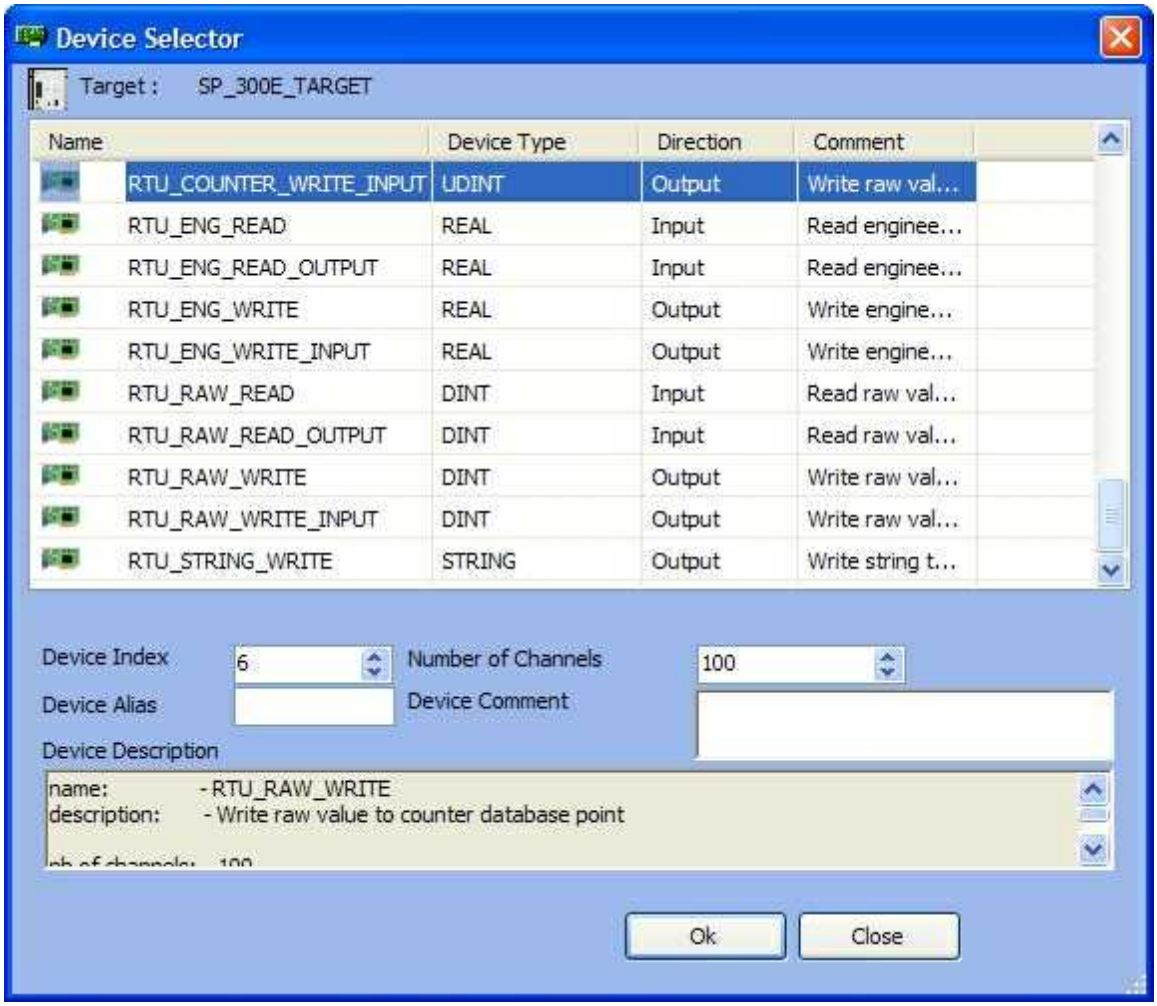
Associated with each device is a ***First_Point_Number*** field that assigns the SCADAPack E database physical counter input point number to the first channel on the device. Variables assigned to subsequent channels are assigned to consecutive point numbers.

Connected variables are updated at the start of the scan with the *Current Integer Value* property of the counter point from the RTU point database.

7.2.6.2 Writing RTU Counter Input Points

RTU_COUNTER_WRITE_INPUT Devices

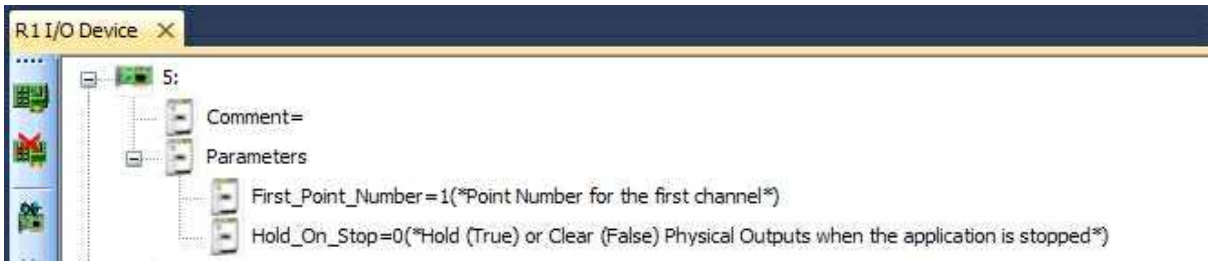
The Channels on this device type are connected to UDINT *VarOutput* variables within an application and write data to points in the RTU point database.



The *RTU_COUNTER_WRITE_INPUT* I/O Device has a user-selectable Number of Channels between 1 and 100.

Associated with each device are the following parameters:

- ***First_Point_Number*** field assigns the SCADAPack E database derived counter point number to the first channel on the device. Subsequent channels are assigned to consecutive database point numbers of the same type
- ***Hold_On_Stop*** field affects RTU point database Physical Output point types only. This parameter has no effect on this I/O device.



RTU point objects supported for output from these devices include Derived counter points & System counter points.

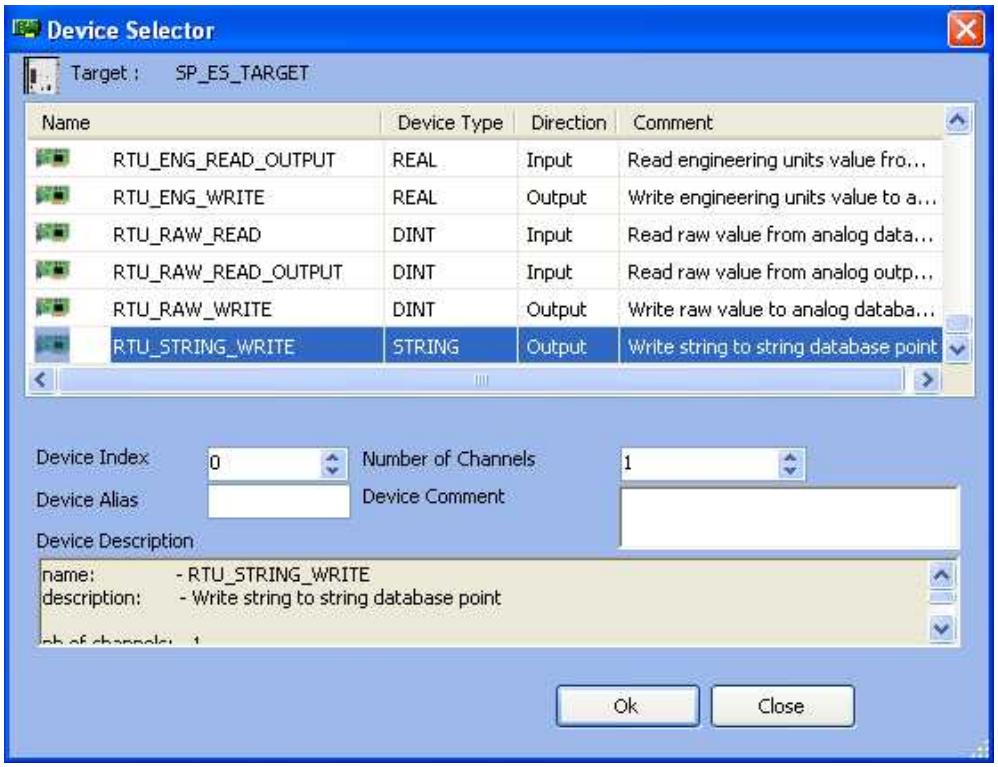
The *Current State* property of the counter input points (in the RTU point database) is updated with the state of the variables at the end of the scan.

RTU database points that correspond to channels on an output device that are not connected to variables are typically set to the OFF state by programs. Even though a point is not connected to a variable on the I/O device, the points do behave as if they are under application control. If it is desired to control the unconnected points external to a resource (e.g. via DNP3, Modbus, IEC60870, etc) reduce the Number of Channels parameter on the device to avoid overlap.

7.2.7 Writing RTU String Points

RTU_STRING_WRITE Devices

The RTU_STRING_WRITE I/O device provides a single string output channel which can be connected to a STRING *VarOutput* variable within an application.



Associated with each device is a ***First_Point_Number*** field that assigns the SCADAPack E database system string point number to the channel on the I/O device.

The variable updates, at the start of the scan, the SCADAPack E database system string point with the value of the STRING variable in the application.

String points are available in the RTU point database as fixed length system string points. For additional

information see *SCADAPack E Configuration Technical Reference* manual.

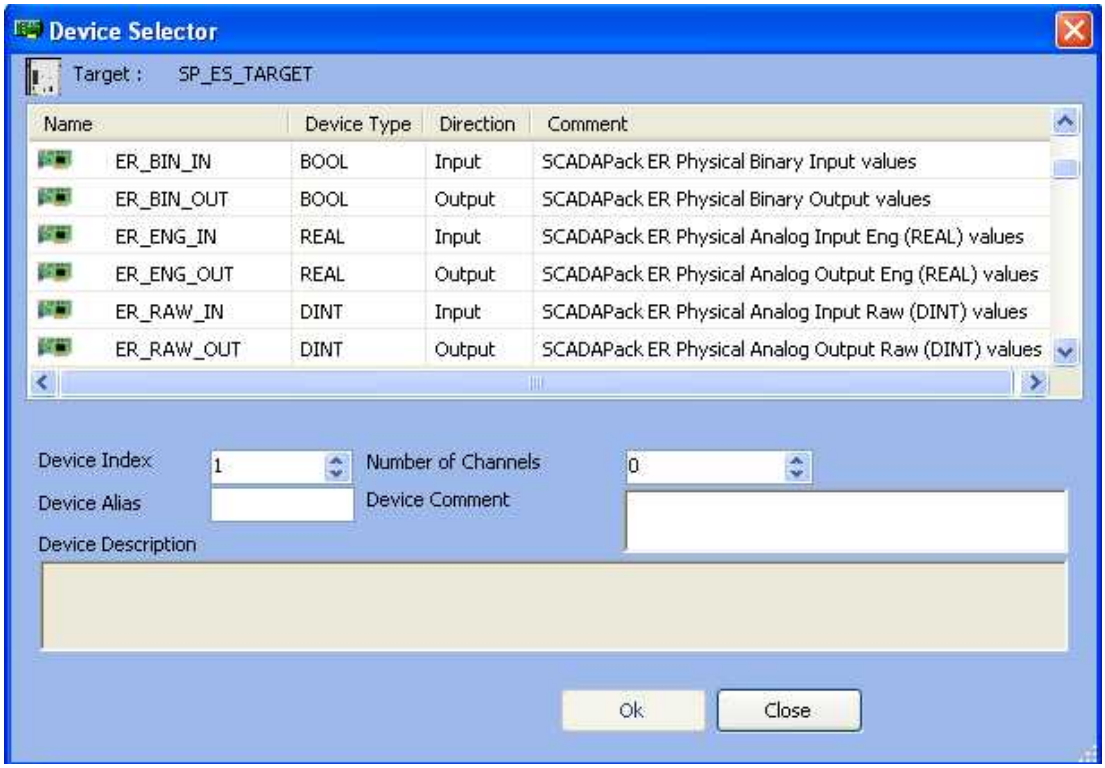
7.2.8 SCADAPack ER I/O Devices

These I/O Devices are only supported on the SCADAPack ER RTUs. Applications containing these device definitions on RTUs other than SCADAPack ER will not start.

The SCADAPack ER I/O devices reference physical channels directly, as opposed to referencing a specific I/O channel by point number. SCADAPack ER I/O devices are listed in the following table:

Device Name	Data Type
ER_BIN_IN	32 BOOL Inputs
ER_BIN_OUT	16 BOOL Outputs
ER_ENG_IN	16 REAL Inputs *
ER_ENG_OUT	4 REAL Outputs *
ER_RAW_IN	16 DINT Inputs *
ER_RAW_OUT	4 DINT Outputs *

* See Analog conversion rules in Section ***Analog I/O Devices / Point Representation & Conversion*** 23



The SCADAPack ER I/O devices reference the respective physical I/O cards by specifying a

Slot_Number field. The *Slot_Number* field is set via user configuration through the I/O Device parameters. These are set as part of the application and are entered into the I/O device parameter fields within the I/O Device Window.

The parameter field requirements are as follows:

Slot_Num: specifies the I/O card slot on the ER rack
1 = I/O Card Slot 1
2 = I/O Card Slot 2, etc.

The default value is 1 (i.e. I/O Card Slot 1).

A valid I/O card configuration needs to be loaded into the SCADAPack ER RTU prior to loading an application that references a SCADAPack ER I/O device, otherwise the I/O device can not be opened. This is done using SCADAPack E Configurator by assigning an I/O card to a rack on and writing the Configurator file changes to the RTU. SCADAPack E Configurator will restart the controller after these configuration details have been written to the RTU. See the *SCADAPack E Configurator User Manual* for details.

7.2.8.1 ER_BIN_IN I/O Device

SCADAPack ER 32 channel digital input device

The ER_BIN_IN input I/O device references a physical digital input card by specifying a *Slot_Number* field (the slot number that the card is installed on a SCADAPack ER RTU rack).

The channel number in the I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Unlike the ER_BIN_OUT output device, there should be RTU database configuration points associated with the physical channels referenced by the ER_BIN_IN input device for proper operation. I.e. SCADAPack E Configurator should be used to configure RTU points on the SCADAPack ER I/O card configuration for the corresponding *Slot_Number*.

Where an application attaches a BOOL *VarInput* variable to an ER_BIN_IN I/O device, the *Current State Property* of the digital point will be read into the variable. The ER_BIN_IN I/O device will successfully open if there is a valid I/O card configuration loaded into the SCADAPack ER controller, and there is at least 1 physical binary input configuration point associated with the given I/O card.

A valid I/O card configuration needs to be loaded into the SCADAPack ER RTU prior to loading an application that references a SCADAPack ER I/O device, otherwise the I/O device can not be opened. This is done using SCADAPack E Configurator by assigning an I/O card to a rack on and writing the Configurator file changes to the RTU. SCADAPack E Configurator will restart the SCADAPack ER after these configuration details have been written. See the *SCADAPack E Configurator User Manual* for details.

7.2.8.2 ER_BIN_OUT I/O Device

SCADAPack ER 16 channel relay output device

The ER_BIN_OUT output I/O device references a physical relay output card by specifying a *Slot_Number* field (the slot number that the card is installed on a SCADAPack ER RTU rack). The channel number in the I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Where an application attaches a BOOL *VarOutput* variable to an ER_BIN_OUT output device, the state of the corresponding output relay will be controlled from the variable. If there is a physical digital output configuration point associated with this physical channel, the *Current State* of this configuration point will be updated after the successful control of the relay output.

Controls issued to SCADAPack ER relay output cards resulting from **attached** variables changing state, are issued as complete I/O card controls. Any simultaneous state changes at the output device level, are executed simultaneously at the SCADAPack ER relay output card.

The ER_BIN_OUT output device may be successfully opened if there is valid I/O card configuration loaded into the SCADAPack ER controller. Unlike standard RTU point output devices (e.g. RTU_BIN_WRITE), it is NOT necessary that there are physical digital output configurations points associated with the physical channels referenced by the ER_BIN_OUT output device.

A valid I/O card configuration needs to be loaded into the SCADAPack ER RTU prior to loading an application that references a SCADAPack ER I/O device, otherwise the I/O device can not be opened. This is done using SCADAPack E Configurator by assigning an I/O card to a rack on and writing the Configurator file changes to the RTU. A controller restart is required after these configuration details have been written to the RTU. See the *SCADAPack E Configurator User Manual* for details.

7.2.8.3 ER_ENG_IN I/O Device

SCADAPack ER 16 channel REAL analog input device

The ER_ENG_IN I/O Device references a physical analog input card by specifying a *Slot_Number* field (the slot number that the card is installed on a SCADAPack ER RTU rack). The channel number in the I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Unlike the ER_ENG_OUT output device, there needs to be point database configuration points associated with the physical channels referenced by the ER_ENG_IN I/O Device for proper operation.

Where an application attaches a REAL (floating point) *VarInput* analog variable to an ER_ENG_IN input device, the *Current Eng Value* property of the associated analog point will be read into the variable. Both DINT and REAL analog variables may **not** be mixed on the same I/O device.

The ER_ENG_IN input device may be successfully opened if there is a valid I/O card configuration loaded into the SCADAPack ER controller, and there is at least 1 physical analog input configuration point associated with the given I/O card.

A valid I/O card configuration needs to be loaded into the SCADAPack ER RTU prior to loading an application that references a SCADAPack ER I/O device, otherwise the I/O device can not be opened. This is done using SCADAPack E Configurator by assigning an I/O card to a rack on and writing the Configurator file changes to the RTU. A controller restart is required after these configuration details have been written to the RTU. See the *SCADAPack E Configurator User Manual* for details.

7.2.8.4 ER_ENG_OUT I/O Device

SCADAPack ER 4 channel REAL analog output device

The ER_ENG_OUT I/O device references a physical analog output card by specifying a *Slot_Number* field (the slot number that the card is installed on a SCADAPack ER RTU rack). The channel number in the I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Where an application attaches an Analog REAL *VarOutput* variable to an ER_ENG_OUT output device, the *Current Eng Value* property of the associated analog point will be controlled from the variable. If there is a physical analog output configuration point associated with this physical channel, the *Current Eng Value* and *Current Integer Value* of this configuration point will be updated after the successful control of the analog output.

The ER_ENG_OUT output device may be successfully opened if there is valid I/O card configuration loaded into the SCADAPack ER controller. Unlike standard RTU point output devices (e.g. RTU_ENG_WRITE), it is NOT necessary that there are physical analog output configurations points associated with the physical channels referenced by the ER_ENG_OUT output device.

A valid I/O card configuration needs to be loaded into the SCADAPack ER RTU prior to loading an application that references a SCADAPack ER I/O device, otherwise the I/O device can not be opened. This is done using SCADAPack E Configurator by assigning an I/O card to a rack on and writing the Configurator file changes to the RTU. A controller restart is required after these configuration details have been written to the RTU. See the *SCADAPack E Configurator User Manual* for details.

7.2.8.5 ER_RAW_IN I/O Device

SCADAPack ER 16 channel DINT analog input device

The ER_RAW_IN I/O device references a physical analog input card by specifying a *Slot_Number* field (the slot number that the card is installed on a SCADAPack ER RTU rack). The channel number in the I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Unlike the ER_RAW_OUT device, there needs to be point database configuration points associated with the physical channels referenced by the ER_RAW_IN input device for proper operation.

Where an application attaches a DINT *VarInput* analog variable to an ER_RAW_IN input device, the *Current Integer Value* property of the associated analog point will be read into the variable.

The ER_RAW_IN input device may be successfully opened if there is a valid I/O card configuration loaded into the SCADAPack ER controller, and there is at least 1 physical analog input configuration point associated with the given I/O card.

A valid I/O card configuration needs to be loaded into the SCADAPack ER RTU prior to loading an application that references a SCADAPack ER I/O device, otherwise the I/O device can not be opened. This is done using SCADAPack E Configurator by assigning an I/O card to a rack on and writing the Configurator file changes to the RTU. A controller restart is required after these configuration details have been written to the RTU. See the *SCADAPack E Configurator User Manual* for details.

7.2.8.6 ER_RAW_OUT I/O Device

SCADAPack ER 4 channel DINT analog output device

The ER_RAW_OUT I/O Device references a physical analog output card by specifying a *Slot_Number* field (the slot number that the card is installed on a SCADAPack ER RTU rack). The channel number in the I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Where an application attaches an Analog "Integer" variable to an ER_RAW_OUT output device, the *Current Integer Value* property of the associated analog point will be controlled from the variable. If there is a physical analog output configuration point associated with this physical channel, the *Current Integer Value* and *Engineering Value* of this configuration point will be updated after the successful control of the relay output.

The ER_RAW_OUT output device may be successfully opened if there is valid I/O card configuration loaded into the SCADAPack ER controller. Unlike standard RTU point output devices (e.g. RTU_RAW_WRITE), it is NOT necessary that there are physical analog output configurations points associated with the physical channels referenced by the ER_RAW_OUT output device.

A valid I/O card configuration needs to be loaded into the SCADAPack ER RTU prior to loading an application that references a SCADAPack ER I/O device, otherwise the I/O device can not be opened. This is done using SCADAPack E Configurator by assigning an I/O card to a rack on and writing the Configurator file changes to the RTU. A controller restart is required after these configuration details have been written to the RTU. See the *SCADAPack E Configurator User Manual* for details.

7.3 Serial Modbus Master I/O Devices

Overview

PLC and peripheral devices may communicate with the Schneider Electric SCADAPack E RTU using PLC Device I/O devices. When a SCADAPack E serial port, or multiple serial ports are configured as *PLC Device* function, the PLC Device I/O devices can communicate with external PLC peripheral devices.

PLC or peripheral device elements are read and the return values cached in the RTU for access through an input device. Similarly, output device data can cause controls to be writtten to the PLC or peripheral device.

The RTU's interface is described in detail in the *SCADAPack E Target 5 Technical Reference* manual. The communications status of transactions between the PLC or peripheral device and the RTU is presented in System Points that can be accessed using variables, or external to the RTU.

A maximum of 200 PLC Device I/O devices (total of each PLC type) may be configured in total per Resource.

When using Modbus PLC I/O devices for communication with a Modbus peripheral device, or devices, the SCADAPack E RTU is a **Modbus Master**. The peripheral device(s) needs to be Modbus Slave(s) and communicate through RTU serial ports configured with the *PLC Device* function.

7.3.1 Modbus RTU I/O Device Types

SCADAPack E RTU can access PLC peripheral through applications by defining I/O device definitions to external PLC or peripheral devices. Standard **RTU** I/O devices can access RTU physical I/O and RTU database points. The I/O devices defined in this section allow data to be extracted from external PLC device(s).

External peripheral data is cached internally by the SCADAPack E RTU to maximize application performance. Access to this cached device data is restricted to resources and is termed PLC Device data.

Access to PLC Device data through the RTU's point database (and then to external communication protocols) requires application code to copy the data (e.g. to variables on **RTU** output devices or using **SETPNTxx** functions, etc).

Modbus RTU Serial

Where a SCADAPack E RTU has one or more of its serial ports configured as '*PLC Device*' and **MBUS..** I/O devices are used, the RTU communicates using serial "MODBUS RTU" protocol.

The SCADAPack E RTU does **not** support "MODBUS ASCII" protocol.

Settings of the RTU communication port such as baud rates and parity format, configured using SCADAPack E Configurator, are used by the RTU's Modbus PLC device driver. RS232, RS422 and RS485 communications are supported.

MBUS_INT_WRITE_SINGLE & **MBUS_UINT_WRITE_SINGLE** I/O devices can **not** be used when multiple communication ports are configured for *PLC Device* function due to the requirement to specify which of these ports connects to the device.

For more information see the *SCADAPack E Target 5 Modbus Communication Interfaces* manual.

The SCADAPack E RTU supports simultaneous communication using serial Modbus and Open Modbus/ TCP or Modbus RTU in TCP protocols. I.e. **MBUS..** I/O devices can communicate with Modbus peripherals on one or more RTU serial ports, and at the same time, **MTCP..** I/O devices can communicate with Modbus/TCP peripherals on the RTU TCP/IP interface (e.g. Ethernet).

Up to a total of 200 PLC Device I/O devices can be defined in total for *PLC Device* communication ports per *Resource*. Multiple SCADAPack E *PLC Device* serial ports, as well as TCP/IP channels, can be used for PLC device peripheral communication.

Communicating with PLCs

LED(s) on the RTU device may indicate communication activity with external peripheral device(s). For more information see relevant SCADAPack E Smart RTU *Hardware User Manual*.

When connecting Workbench Debugger to a SCADAPack E Smart RTU using PLC Device I/O devices, the Debugger may indicate "DISCONNECTED" for a period of time, particularly if there is a large number of PLC Device I/O devices, or if a PLC is not responding. In this case please wait. The connection will be established after the application works through the PLC device updates.

PLC Data

Different PLC Device I/O devices are provided for different types of PLC data. For example: analog DINT_READ & REAL_READ devices are provided to read PLC value registers, BOOL_WRITE devices for writing to PLC coils and analog DINT_WRITE & REAL_WRITE devices to read PLC accumulated data. The different types of I/O devices available and ranges of PLC data that can be accessed depend on the individual PLC driver.

PLC Device I/O devices access Modbus PLC data in the following way:

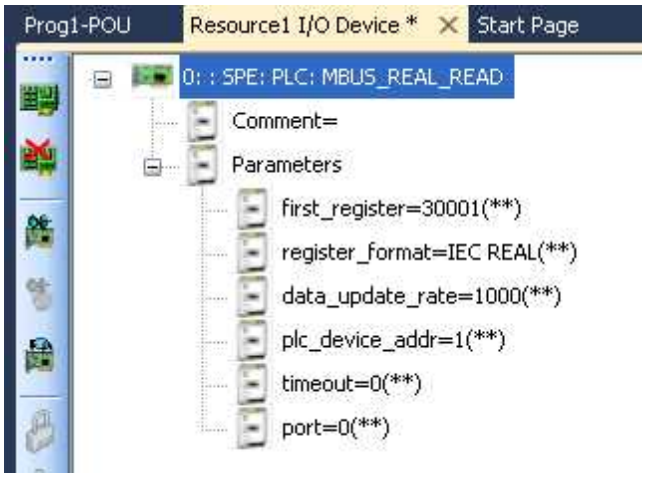
- a PLC Device READ device reads PLC data from read-only and read-write registers
- a PLC Device WRITE device writes PLC data to read-write registers
- Serial communication with external devices, such as PLC's, is made through the RTU port(s) configured as *PLC Device*.

I/O Devices are *strongly-typed*. Both DINT and REAL analog variables may **not** be mixed on the same Analog I/O device. However, conversion functions such as ANY_TO_DINT or ANY_TO_REAL may be used to perform a data type conversion in the user application as necessary.

7.3.2 Reading Modbus Registers

PLC I/O input devices typically require user configuration through the I/O device parameters. These are set as part of the application and are entered into the I/O device parameter fields within the *I/O Device* Window.

The SCADAPack E RTU can communicate with Modbus PLC peripheral devices via the serial 'MODBUS RTU', Open Modbus/TCP or Modbus RTU in TCP protocols. For serial Modbus connections to the peripheral PLC device, the I/O device types with prefix **MBUS** should be used.



Typical fields are:

***first_register**:* specifies the PLC Device data registers to access when reading PLC data into variables. The PLC data type accessed is specific to the PLC Device I/O device and address. This value is usually the PLC's data (or register) address.

***register_format**:* specifies the PLC data register type. The following data types are supported: IEC DISCRETE, 984 DISCRETE, IEC UINT IEC INT, IEC DINT, IEC REAL, SWAP REAL. See the *SCADAPack E Target 5 Modbus Communication Interfaces* manual for more information.

***data_update_rate**:* The units for this parameter vary depending on the type of PLC device. For example this may be a setting in milliseconds for a directly connected device, or in minutes for a low power type device (see the *SCADAPack E Target 5 Modbus Communication Interfaces* manual). As the SCADAPack E RTU needs to extract the data for the I/O device from the PLC or peripheral device, this sets the rate at which the data is extracted. Individual I/O devices may have different data update rates allowing prioritization of data extracted from a PLC device. The RTU may not be able to read requested PLC data within the time set by the data update rate depending on the quantity of data to be read, rate of write requests and PLC communication speed. In this case the update rates will be slower. For **MTCP** I/O device types, the time specified in this field is in milliseconds.

***plc_device_addr**:* This parameter specified the PLC device address. Some PLC device drivers support multi-drop PLC devices on the same communication channel, or have unique addressing identifiers. Where the RTU driver provides multi-drop support, Resources may access data from any of the locally multi-dropped devices. A separate I/O device will be required for each device.

***timeout**:* PLC device drivers with comprehensive I/O device interfaces may provide a parameter for specifying the communications timeout on an individual I/O device (i.e. the timeout applies to communications associated with that device). Where this value is "0", the PLC device driver will use a default timeout of 1200 milliseconds. The units for this field are dependent upon the PLC device driver. Units may be, for example, milliseconds, seconds, minutes, etc. For **MTCP** I/O device types, the unit of

this field is in milliseconds.

***Port**:* This parameter is only available on the serial Modbus I/O device driver with the **MBUS** prefix. Where present, it defines which of multiple RTU *PLC Device* ports will be used to communicate with the PLC or peripheral device. PLC Device I/O devices not including this parameter can only be used when a single *PLC Device* port is configured on the SCADAPack E RTU.

7.3.2.1 MBUS_BOOL_READ I/O Device

Serial Modbus PLC 32 channel BOOL input device

Description

The MBUS_BOOL_READ I/O device provides up to 32 digital input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device via a serial connection. PLC data supported include coil, digital input status and holding registers. Each connected BOOL variable is updated continuously with the current value of the Modbus register. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td>
<table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>1 - 9999</td><td>Read Coils - Modbus Function Code 1</td></tr>
<tr><td>10001 - 19999</td><td>Read Status Register - Modbus Function Code 2</td></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table>
</td></tr>
<tr><td>register_format</td><td>IEC DISCRETE, 984 DISCRETE</td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus slave address (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>Port</td><td>0 = port0 (SCADAPack ES/SCADAPack ER only)
1 = port1
2 = port2
3 = port3
4 = port4 (SCADAPack ES/SCADAPack ER only)</td></tr>
</table>

7.3.2.2 MBUS_DINT_READ I/O Device

Serial Modbus PLC 62 Channel DINT Input device

Description

The MBUS_DINT_READ I/O device provides up to 62 IEC DINT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device with a serial connection. PLC data supported includes inputs, and holding registers. Each connected DINT variable is updated continuously with the current value of two consecutive Modbus registers. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65534</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus slave Address (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>Port</td><td>0 = port0 (SCADAPack ES/SCADAPack ER only)
1 = port1
2 = port2
3 = port3
4 = port4 (SCADAPack ES/SCADAPack ER only)</td></tr>
</table>

7.3.2.3 MBUS_INT_READ I/O Device

Serial Modbus PLC 124 Channel INT Input device

Description

The MBUS_INT_READ I/O device provides up to 124 IEC INT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device with a serial connection. PLC data supported includes inputs, and holding registers. Each connected INT variable is updated continuously with the current value of the Modbus register. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus slave Address (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>Port</td><td>0 = port0 (SCADAPack ES/SCADAPack ER only)
1 = port1
2 = port2
3 = port3
4 = port4 (SCADAPack ES/SCADAPack ER only)</td></tr>
</table>

7.3.2.4 MBUS_REAL_READ I/O Device

Serial Modbus PLC 62 Channel REAL Input device

Description

The MBUS_REAL_READ I/O device provides up to 62 IEC REAL analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device with a serial connection. PLC data supported includes inputs, and holding registers. Each connected REAL variable is updated continuously with the current value of two consecutive Modbus registers. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65534</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>register_format</td><td>IEC REAL, SWAP REAL</td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus slave Address (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>Port</td><td>0 = port0 (SCADAPack ES/SCADAPack ER only)
1 = port1
2 = port2
3 = port3
4 = port4 (SCADAPack ES/SCADAPack ER only)</td></tr>
</table>

7.3.2.5 MBUS_UINT_READ I/O Device

Serial Modbus PLC 124 Channel UINT Input device

Description

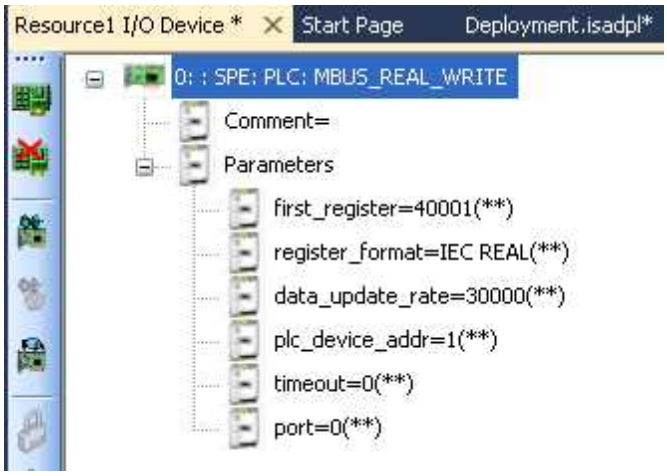
The MBUS_INT_READ I/O device provides up to 124 IEC UINT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device with a serial connection. PLC data supported includes inputs, and holding registers. Each connected UINT variable is updated continuously with the current value of the Modbus register. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus slave Address (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>Port</td><td>0 = port0 (SCADAPack ES/SCADAPack ER only)
1 = port1
2 = port2
3 = port3
4 = port4 (SCADAPack ES/SCADAPack ER only)</td></tr>
</table>

7.3.3 Writing Modbus Registers

PLC Device output devices require user configuration through the I/O device parameters. These are set as part of the application and are entered into the I/O device parameter fields within the *I/O Device* Window..



The device to which Modbus Output commands are sent needs to provide Modbus register addresses for each of the channels on the output device, regardless of whether variables are attached to the channels, or not.

E.g. for the MBUS_BIN_WRITE device, 16 contiguous Modbus Coil registers needs to be present in the remote device and support external writing.

The Device Parameter fields for Modbus Output devices are:

first_register*:* specifies the PLC Device data registers to access when writing from variables to PLC data. The PLC data type accessed is specific to the PLC Device I/O device and address. This value is usually the PLC's data (or register) address.

register_format*:* specifies the PLC data register type. Currently *IEC UINT, INT, DINT, IEC REAL & SWAP REAL* types are supported for analog devices and *IEC DISCRETE* type is supported for Boolean devices. See the *SCADAPack E Target 5 Modbus Communication Interfaces* manual for more information.

plc_device_addr*:* Some PLC device drivers support multi-drop PLC devices on the same communication channel, or have unique addressing identifiers. Where the RTU driver provides multi-drop support, resources may access data from any of the locally multi-dropped devices. A separate I/O device will be required for each PLC device.

data_update_rate*:* The unit for this parameter (when present on Modbus Output devices) is in milliseconds, and configures the rate at which the data for the Output device is written to the PLC. Between "data_update_*rate*" periods, data is written to the PLC only when the output channel variable values change. Individual I/O devices may have different data update rates allowing prioritization of data sent to a PLC device. Setting this parameter to 0 disables the time-based writing of output data. Data is written at application startup (every channel) and thereafter only when individual output channel variables change.

***timeout**:* PLC device drivers with comprehensive I/O device interfaces may provide a parameter for specifying the communications timeout on an individual I/O device (i.e. the timeout applies to communications associated with that device). Where this value is “0”, the PLC device driver will use a default timeout. The units for this field are dependent upon the PLC device driver. Units may be, for example, milliseconds, seconds, minutes, etc.

port: this parameter may be on a PLC Device I/O device for a device driver. Where present, it defines which of multiple RTU *PLC Device* ports will be used to communicate with the PLC or peripheral device. If only one *PLC Device* port is configured, this field is ignored. PLC Device I/O devices not including this parameter can only be used when a single *PLC Device* port is configured on the SCADAPack E RTU.

7.3.3.1 MBUS_BOOL_WRITE I/O Device

Serial Modbus PLC 32 channel BOOL Output device

The MBUS_BOOL_WRITE I/O device provides up to 32 digital output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device. The digital output channels can be tied to BOOL *VarOutput* variables within an application. PLC data supported include relays coils and holding registers. A target Modbus register is updated as necessary for each of the connected BOOL output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>1 - 9999</td><td>Write Coil - Modbus Function Code 5</td></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
register_format	IEC DISCRETE, 984 DISCRETE
data_update_rate	units in milliseconds
plc_device_addr	Modbus slave address (1 - 254)
Timeout	units in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)

7.3.3.2 MBUS_DINT_WRITE I/O Device

Serial Modbus PLC 61 channel DINT Output device

The MBUS_DINT_WRITE I/O device provides up to 61 IEC DINT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device via a serial connection. The analog output channels can be tied to DINT *VarOutput* variables within an application. PLC data supported include outputs and holding registers. Two consecutive Modbus registers are updated as necessary for each of the connected DINT output variables.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65534</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus slave address (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>Port</td><td>0 = port0 (SCADAPack ES/SCADAPack ER only)
1 = port1
2 = port2
3 = port3
4 = port4 (SCADAPack ES/SCADAPack ER only)</td></tr>
</table>

7.3.3.3 MBUS_INT_WRITE I/O Device

Serial Modbus PLC 122 channel INT Output device

The MBUS_INT_WRITE I/O device provides up to 122 IEC INT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device via a serial connection. The analog output channels can be tied to INT *VarOutput* variables within an application. PLC data supported include outputs and holding registers. A target Modbus register is updated as necessary for each of the connected INT output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
data_update_rate	units in milliseconds
plc_device_addr	Modbus slave address (1 - 254)
Timeout	units in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)

7.3.3.4 MBUS_INT_WRITE_SINGLE I/O Device

Serial Modbus PLC 1 channel INT Output device

The MBUS_INT_WRITE_SINGLE I/O device provides 1 IEC INT analog output channel for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device via a serial connection. The analog output channel can be tied to an INT *VarOutput* variable within an application. PLC data supported include holding registers. The connected I/O points are updated continuously with the *Current Value* of the output variable. This information is cached internally by the SCADAPack E RTU and made available to the I/O points.

These devices differ from the MBUS_xxx_WRITE devices. The device does not provide the flexibility of specifying data types, data update rate, timeout or RTU port number. This writes a single 16-bit value to a Modbus Holding Register (in IEC INT format). Registers are only written to the PLC peripheral device if they change. Modbus Function Code 6 (FC 6) is requested. A fixed timeout of 1200 milliseconds is used.

This device can only be used if a single RTU port is configured as a *PLC Device* function. As the device does not specify a port to use, its operation is not supported if there are multiple ports configured as *PLC Device* function.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 6</td></tr></table></td></tr>
<tr><td>plc_device_addr</td><td>Modbus slave address (1 - 254)</td></tr>
</table>

7.3.3.5 MBUS_REAL_WRITE I/O Device

Serial Modbus PLC 61 channel REAL Output device

The MBUS_REAL_WRITE I/O device provides up to 61 IEC REAL analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device via a serial connection. The analog output channels can be tied to REAL *VarOutput* variables within an application. PLC data supported include outputs and holding registers. Two consecutive Modbus registers are updated as necessary for each of the connected REAL output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65534</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
register_format	IEC REAL, SWAP REAL
data_update_rate	units in milliseconds
plc_device_addr	Modbus slave address (1 - 254)
Timeout	units in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)

7.3.3.6 MBUS_UINT_WRITE I/O Device

Serial Modbus PLC 122 channel UINT Output device

The MBUS_UINT_WRITE I/O device provides up to 122 IEC UINT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device via a serial connection. The analog output channels can be tied to UINT *VarOutput* variables within an application. PLC data supported include outputs and holding registers. A target Modbus register is updated as necessary for each of the connected UINT output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
data_update_rate	units in milliseconds
plc_device_addr	Modbus slave address (1 - 254)
Timeout	units in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)

7.3.3.7 MBUS_UINT_WRITE_SINGLE I/O Device

Serial Modbus PLC 1 channel UINT Output device

The MBUS_UINT_WRITE_SINGLE I/O device provides 1 IEC UINT analog output channel for a SCADAPack E RTU to communicate with a Modbus PLC peripheral device via a serial connection. The analog output channel can be tied to an UINT *VarOutput* variable within an application. PLC data supported include holding registers. The connected I/O points are updated continuously with the *Current Value* of the output variable. This information is cached internally by the SCADAPack E RTU and made available to the I/O points.

These devices differ from the MBUS_xxx_WRITE devices. The device does not provide the flexibility of specifying data types, data update rate, timeout or RTU port number. This writes a single 16-bit value to a Modbus Holding Register (in IEC UINT format). Registers are only written to the PLC peripheral device if they change. Modbus Function Code 6 (FC 6) is requested. A fixed timeout of 1200 milliseconds is used.

This device can only be used if a single RTU port is configured as a *PLC Device* function. As the device does not specify a port to use, its operation is not supported if there are multiple ports configured as *PLC Device* function.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 6</td></tr></table></td></tr>
<tr><td>plc_device_addr</td><td>Modbus slave address (1 - 254)</td></tr>
</table>

7.4 Modbus TCP Client I/O Devices

For Open Modbus/TCP connections to the peripheral device, the I/O device types with prefix **MTCP** should be used.

The **MTCP..** devices allow Ethernet or PPP serial interfaces to be used by an SCADAPack E RTU to communicate with Open Modbus/TCP protocol peripheral devices

Each of the application's PLC I/O devices use a separate Modbus/TCP request to read or write its data. Improved Modbus communication efficiency can be achieved by grouping Modbus registers together and using less I/O devices with a larger number of channels, rather than more I/O devices with a smaller number of channels.

A maximum of 200 PLC Device I/O devices (total of every PLC type) may be configured in total per Resource.

When using Modbus/TCP I/O devices for communication with Modbus/TCP peripheral devices, the SCADAPack E RTU is an **Open Modbus/TCP Client**. The peripheral device(s) needs to be Open Modbus/TCP **Server**(s) (e.g. Ethernet PLC). Open Modbus/TCP protocol is also known as MBAP protocol, but is referred to as Open Modbus/TCP protocol throughout this manual.

Modbus/TCP devices utilize default IEC data types. Where applicable, the data type may be available for the user to choose.

7.4.1 Modbus/TCP I/O Device Types

SCADAPack E RTU can access PLC peripheral through applications by defining I/O device definitions to external PLC or peripheral devices. Standard I/O devices can access RTU physical I/O and RTU database points. The I/O devices defined in this section allow data to be extracted from external PLC device(s).

External peripheral data is cached internally by the SCADAPack E RTU to maximize application performance. Access to this cached device data is restricted to resources and is termed PLC Device data.

Access to PLC Device data through the RTU's point database (and then to external communication protocols) requires application code to copy the data (e.g. to variables on RTU_xxx_WRITE devices, or using SETPNTxx functions, etc).

Modbus/TCP

SCADAPack E RTU's, with *Modbus/IP (Client)* service enabled and when using **MTCP..** I/O devices, communicate using Open Modbus/TCP communication protocol. TCP Port number 502 is normally used for connection to the PLC peripheral device for Modbus/TCP communications.

The protocol connects TCP socket(s) between the SCADAPack E RTU (Client) and the peripheral device (s) (Servers). TCP/IP over Ethernet and PPP interfaces from the SCADAPack E RTU are supported.

For more information see the *SCADAPack E Target 5 Modbus Communication Interfaces* manual.

The SCADAPack E RTU supports simultaneous communication using serial Modbus and Open Modbus/ TCP protocols. I.e. **MBUS..** I/O devices can communicate with Modbus peripherals on one or more RTU serial ports, and at the same time, **MTCP..** I/O devices can communicate with Modbus/TCP peripherals on the RTU TCP/IP interface (e.g. Ethernet).

Communicating with PLCs

TCP/IP, serial port or Ethernet LED(s) on the RTU device may indicate communication activity with external peripheral Modbus/TCP device(s). For more information see relevant SCADAPack E Smart RTU *Hardware User Manual*.

When connecting Workbench Debugger to a SCADAPack E RTU using PLC Device I/O devices, the Debugger may indicate "DISCONNECTED" for a period of time, particularly if there is a large number of PLC Device I/O devices, or if a PLC is not responding. In this case please wait. The connection will be established after the application works through the PLC device updates.

PLC Data

Different PLC Device I/O devices are provided for different types of PLC data. For example: analog DINT_READ & REAL_READ devices are provided to read PLC value registers, BOOL_WRITE devices for writing to PLC coils and analog DINT_WRITE & REAL_WRITE devices to read PLC accumulated data. The different types of I/O devices available and ranges of PLC data that can be accessed depend on the individual PLC driver.

PLC Device I/O devices access Modbus PLC data in the following way:

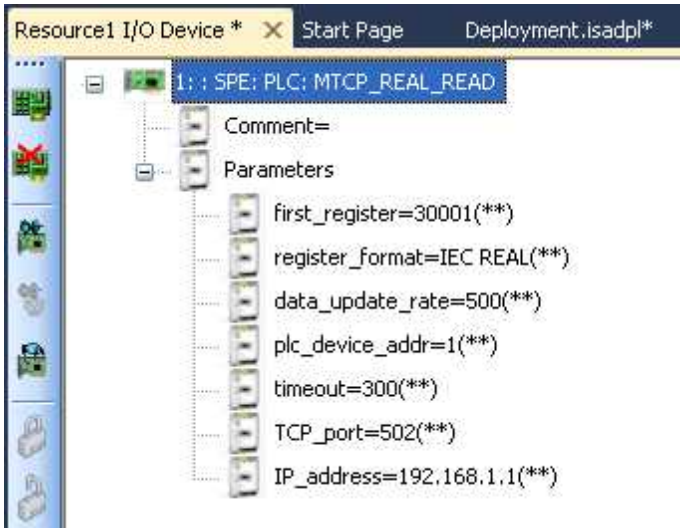
- a PLC Device READ device reads PLC data from read-only and read-write registers
- a PLC Device WRITE device writes PLC data to read-write registers

Up to a total of 200 PLC Device I/O devices can be defined in total for *PLC Device* communication ports per *Resource*. Multiple SCADAPack E *PLC Device* serial ports, as well as TCP/IP channels, can be used for PLC device peripheral communication.

I/O Devices are *strongly-typed*. Both DINT and REAL analog variables may **not** be mixed on the same Analog I/O device. However, conversion functions such as ANY_TO_DINT or ANY_TO_REAL may be used to perform a data type conversion in the user application as necessary.

7.4.2 Reading Modbus/TCP Registers

Modbus/TCP PLC Input device variables are updated at the start of an application scan. The value presented to the variables is the value returned by the PLC to the previous read request. This read may have occurred during previous application scans. The “data update rate” parameter on the I/O device sets the "scan" rate of the PLC data. The PLC communication status is updated if there is a status code returned from the PLC, or no response from the PLC after a data request by the SCADAPack E RTU. The status is cleared by the RTU upon successful communications. To catch transient status codes you can use code to store non-zero values.



Input Device Parameters

first_register*:* specifies the Modbus/TCP PLC data registers to access when reading from PLC data into variables. The PLC data type accessed is the same as Modbus PLC Device I/O devices detailed in Section ***Serial Modbus Master I/O Devices*** 47.

register_format: specifies the Modbus/TCP PLC data register type. The various types supported include IEC DISCRETE, 984 DISCRETE, IEC UINT, IEC INT, IEC DINT, IEC REAL and SWAP REAL. See the *SCADAPack E Target 5 Modbus Communication Interfaces* manual for more information.

data_update_rate*:* The units for this parameter are set in Milliseconds, and specify the rate at which the data for the Input device is extracted from the PLC. Individual I/O devices may have different data update rates allowing prioritization of data extracted from a PLC device. The SCADAPack E RTU may not be able to read requested PLC data within the time set by the data update rate depending on the quantity of data to be read, rate of write requests and PLC communication speed. In this case the update rates will be slower.

plc_device_address*:* This parameter specifies the PLC device (unit) address. Modbus PLC devices accessed at the same IP address (e.g. via a Modbus bridge) needs to have a unique Unit address in order to be identified. Resources may access data from different units on the same IP address or at different IP addresses. In these cases a separate I/O device will be required for each device.

timeout: The Modbus/TCP PLC device driver provides a parameter for specifying the communications timeout on an individual I/O device (i.e. the timeout applies to communications associated with that device). Where this value is “0”, the PLC device driver will use the default timeout (1200 milliseconds). Units for this field are in milliseconds.

tcp_port: Default TCP/IP port number = 502

IP_address: This parameter specifies the IP network address that the SCADAPack E RTU connects to for communication with the PLC for this I/O device. Enter the IP address of the Modbus/TCP PLC, or Modbus Bridge if applicable.

7.4.2.1 MTCP_BOOL_READ I/O Device

Open Modbus/TCP PLC 32 channel BOOL input device

Description

The MTCP_BOOL_READ I/O device provides up to 32 IEC BOOL input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The digital channels can be connected to BOOL *VarInput* variables within an application. PLC data supported include coils, digital input status and holding registers. Each connected BOOL variable is updated continuously with the current value of the Modbus register. This information is cached internally by the RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To ...</th></tr>
<tr><td>1 - 9999</td><td>Read Coils - Modbus Function Code 1</td></tr>
<tr><td>10001 - 19999</td><td>Read Status Register - Modbus Function Code 2</td></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>plc_data_type</td><td>IEC DISCRETE, 984 DISCRETE</td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Default = 502</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.4.2.2 MTCP_DINT_READ I/O Devices

Open Modbus/TCP PLC 62 Channel DINT Input device

Description

The MTCP_DINT_READ I/O device provides up to 62 IEC DINT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The analog input channels can be connected to DINT *VarInput* variables within an application. PLC data supported include inputs, and holding registers. Each connected DINT variable is updated continuously with the current value of two consecutive Modbus registers. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65534</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Default = 502</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.4.2.3 MTCP_INT_READ I/O Devices

Open Modbus/TCP PLC 124 Channel INT Input device

Description

The MTCP_INT_READ I/O device provides up to 124 IEC INT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The analog input channels can be connected to INT *VarInput* variables within an application. PLC data supported include inputs, and holding registers. Each connected INT variable is updated continuously with the current value of the Modbus register. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Default = 502</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.4.2.4 MTCP_REAL_READ I/O Devices

Open Modbus/TCP PLC 62 Channel REAL Input device

Description

The MTCP_REAL_READ I/O device provides up to 62 IEC REAL analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The analog input channels can be connected to REAL *VarInput* variables within an application. PLC data supported include inputs, and holding registers. Each connected REAL variable is updated continuously with the current value of two consecutive Modbus registers. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65534</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>register_format</td><td>IEC REAL, SWAP REAL</td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Default = 502</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.4.2.5 MTCP_UINT_READ I/O Devices

Open Modbus/TCP PLC 124 Channel UINT Input device

Description

The MTCP_UINT_READ I/O device provides up to 124 IEC UINT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The analog input channels can be connected to UINT *VarInput* variables within an application. PLC data supported include inputs, and holding registers. Each connected UINT variable is updated continuously with the current value of the Modbus register. This information is cached internally by the SCADAPack E RTU and made available to the application.

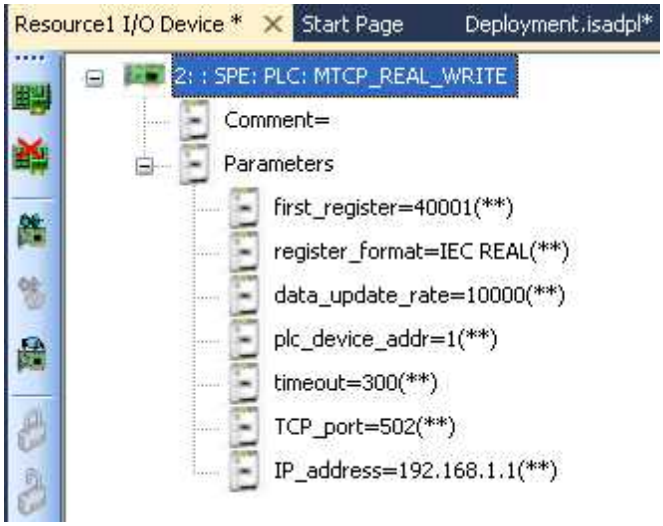
I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Default = 502</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.4.3 Writing Modbus/TCP Registers

Modbus/TCP PLC Output device data is updated to the PLC when an application changes the value of a variable attached to the Output device. In addition, output device data is updated to the PLC under the following conditions:

- When the application starts output device data is written
- If the PLC does not respond to a control, it is re-sent until it is responded



- Output device data is rewritten at the background *data_update_rate* update rate (unless the *data_update_rate* parameter is set to 0)

The device to which Modbus Output commands are sent needs to provide Modbus register addresses for every the channel on the output device, regardless of whether variables are attached to the channels, or not.

E.g. for the MTCP_BIN_WRITE device, 16 contiguous Modbus Coil registers needs to be present in the remote device and support external writing.

Output Device Parameters

first_register*:* specifies the Modbus/TCP PLC data registers to access when reading from PLC data into variables. The PLC data type accessed is the same as Modbus PLC Device I/O devices detailed in Section ***Output Devices*** 47.

register_format*:* specifies the Modbus/TCP PLC data register type.The various types supported include IEC DISCRETE, 984 DISCRETE, IEC UINT, IEC INT, IEC DINT, IEC REAL and SWAP REAL. See the *SCADAPack E Target 5 Modbus Communication Interfaces* manual for more information.

plc_device_address*:* This parameter specifies the PLC device (unit) address. Modbus PLC devices accessed at the same IP address (e.g. via a Modbus bridge) needs to have a unique unit address in order to be identified. Resources may access data from different units on the same IP address or at

different IP addresses. In these cases a separate I/O device will be required for each device.

***data_update_rate**:* The unit for this parameter is set in Milliseconds, and specifies the rate at which the data for the Output device is written to the PLC. Between "*data_update_rate*" periods, data is only written to the PLC when the output variable values change. Individual I/O devices may have different data update rates allowing prioritization of data sent to PLC devices. Setting this parameter to 0 disables the time-based writing of output data. Data is written at application startup (every channel) and thereafter only when individual output channel variables change.

***timeout**:* The Modbus/TCP PLC device driver provides a parameter for specifying the communications timeout on an individual I/O device (i.e. the timeout applies to communications associated with that device). Where this value is "0", the PLC device driver will use the default timeout (1200 milliseconds). Units for this field are in milliseconds.

tcp_port: Default TCP/IP port number = 502

***IP_address**:* This parameter specifies the IP network address that the RTU connects to for communication with the PLC for this I/O device. Enter the IP address of the Modbus/TCP PLC, or Modbus Gateway or Modbus Bridge, as applicable.

7.4.3.1 MTCP_BOOL_WRITE I/O Device

Open Modbus/TCP PLC 32 channel BOOL output device

Description

The MTCP_BOOL_WRITE I/O device provides up to 32 IEC BOOL output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The digital output channel can be tied to BOOL VarOutput variables within an application. PLC data supported include relays coils and holding registers. A target Modbus register is updated as necessary for each of the connected BOOL output variables.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>1 - 9999</td><td>Write Coil - Modbus Function Code 5</td></tr>
<tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr>
</table></td></tr>
<tr><td>plc_data_type</td><td>IEC DISCRETE, 984 DISCRETE</td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Default = 502</td></tr>
<tr><td>IP_address</td><td>IP address of PLC device (e.g. 172.123.250.104)</td></tr>
</table>

7.4.3.2 MTCP_DINT_WRITE I/O Device

Open Modbus/TCP PLC 61 channel DINT output device

Description

The MTCP_DINT_WRITE I/O device provides up to 61 IEC DINT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The analog output channels can be tied to DINT *VarOutput* variables within an application. PLC data supported include analog outputs and holding registers. Two consecutive Modbus registers are updated as necessary for each of the connected DINT output variables.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65534</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Default = 502</td></tr>
<tr><td>IP_address</td><td>IP address of PLC device (e.g. 172.123.250.104)</td></tr>
</table>

7.4.3.3 MTCP_INT_WRITE I/O Device

Open Modbus/TCP PLC 122 channel INT output device

Description

The MTCP_INT_WRITE I/O device provides up to 122 IEC INT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The analog output channels can be tied to INT *VarOutput* variables within an application. PLC data supported include analog outputs and holding registers. A target Modbus register is updated as necessary for each of the connected INT output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
data_update_rate	units in milliseconds
plc_device_addr	Modbus Unit Identifier (1 - 254)
Timeout	units in milliseconds
TCP_port	Default = 502
IP_address	IP address of PLC device (e.g. 172.123.250.104)

7.4.3.4 MTCP_REAL_WRITE I/O Device

Open Modbus/TCP PLC 61 channel REAL output device

Description

The MTCP_REAL_WRITE I/O device provides up to 61 IEC REAL analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The analog output channels can be tied to REAL *VarOutput* variables within an application. PLC data supported include analog outputs and holding registers. Two consecutive Modbus registers are updated as necessary for each of the connected REAL output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65534</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
register_format	IEC REAL, SWAP REAL
data_update_rate	units in milliseconds
plc_device_addr	Modbus Unit Identifier (1 - 254)
Timeout	units in milliseconds
TCP_port	Default = 502
IP_address	IP address of the peripheral PLC device (e.g. 172.244.199.200)

7.4.3.5 MTCP_UINT_WRITE I/O Device

Open Modbus/TCP PLC 122 channel UINT output device

Description

The MTCP_UINT_WRITE I/O device provides up to 122 IEC UINT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The analog output channels can be tied to UINT *VarOutput* variables within an application. PLC data supported include analog outputs and holding registers. A target Modbus register is updated as necessary for each of the connected UINT output variables.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>Timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Default = 502</td></tr>
<tr><td>IP_address</td><td>IP address of PLC device (e.g. 172.123.250.104)</td></tr>
</table>

7.5 Modbus RTU in TCP Client I/O Devices

For Modbus RTU in TCP connections to the peripheral device, the I/O device types with prefix **MRTUTCP** should be used.

The **MRTUTCP..** I/O devices allow Ethernet or PPP serial interfaces to be used by an SCADAPack E RTU to communicate with Modbus RTU in TCP protocol peripheral devices

Each of the application's PLC I/O devices use a separate Modbus RTU in TCP request to read or write its data. Improved Modbus communication efficiency can be achieved by grouping Modbus registers together and using less I/O devices with a larger number of channels, rather than more I/O devices with a smaller number of channels.

A maximum of 200 PLC Device I/O devices (total of every PLC type) may be configured in total per Resource.

When using Modbus RTU in TCP I/O devices for communication with Modbus RTU in TCP peripheral devices, the SCADAPack E RTU is an **Modbus RTU in TCP Client**. The peripheral device(s) needs to be **Modbus RTU in TCP Server**(s) (e.g. Ethernet PLC).

Modbus RTU in TCP devices utilize default IEC data types. Where applicable, the data type may be available for the user to choose.

- ***Modbus RTU in TCP I/O Device Types*** 84
- ***Reading Modbus RTU in TCP Registers*** 85
- ***Reading Modbus RTU in TCP Registers*** 92

7.5.1 Modbus RTU in TCP I/O Device Types

SCADAPack E RTU can access PLC peripheral through applications by defining I/O device definitions to external PLC or peripheral devices. Standard I/O devices can access RTU physical I/O and RTU database points. The I/O devices defined in this section allow data to be extracted from external PLC device(s).

External peripheral data is cached internally by the SCADAPack E RTU to maximize application performance. Access to this cached device data is restricted to Resources and is termed PLC Device data.

Access to PLC Device data through the RTU's point database (and then to external communication protocols) requires application code to copy the data (e.g. to variables on RTU_xxx_WRITE devices, or using SETPNTxx functions, etc).

Modbus RTU in TCP

SCADAPack E RTU's, with *Modbus/IP(client)* service enabled and when using **MRTUTCP..** I/O devices, communicate using Modbus RTU in TCP communication protocol. TCP Port number 49152 is commonly used for connection to the PLC peripheral device for Modbus RTU in TCP communications. The protocol connects TCP socket(s) between the SCADAPack E RTU (Client) and the peripheral device (s) (Servers).

For more information see the *SCADAPack E Target 5 Modbus Communication Interfaces* manual.

The SCADAPack E RTU supports simultaneous communication using serial Modbus and Modbus RTU in TCP protocols. I.e. **MBUS..** I/O devices can communicate with Modbus peripherals on one or more RTU serial ports, and at the same time, **MRTUTCP..** I/O devices can communicate with Modbus RTU in TCP peripherals on the RTU TCP/IP interface (e.g. Ethernet).

Up to a total of 200 PLC Device I/O devices can be defined in total for *PLC Device* communication ports per *Resource*. Multiple SCADAPack E *PLC Device* serial ports, as well as TCP/IP channels, can be used for PLC device peripheral communication.

Communicating with PLCs

TCP/IP, serial port or Ethernet LED(s) on the RTU device may indicate communication activity with external peripheral Modbus RTU in TCP device(s). For more information see relevant SCADAPack E Smart RTU *Hardware User Manual*.

When connecting Workbench Debugger to a SCADAPack E RTU using PLC Device I/O devices, the Debugger may indicate "DISCONNECTED" for a period of time, particularly if there is a large number of PLC Device I/O devices, or if a PLC is not responding. In this case please wait. The connection will be established after the application works through the PLC device updates.

PLC Data

Different PLC Device I/O devices are provided for different types of PLC data. For example: analog DINT_READ & REAL_READ devices are provided to read PLC value registers, BOOL_WRITE devices for writing to PLC coils and analog DINT_WRITE & REAL_WRITE devices to read PLC accumulated data. The different types of I/O devices available and ranges of PLC data that can be accessed depend on the

individual PLC driver.

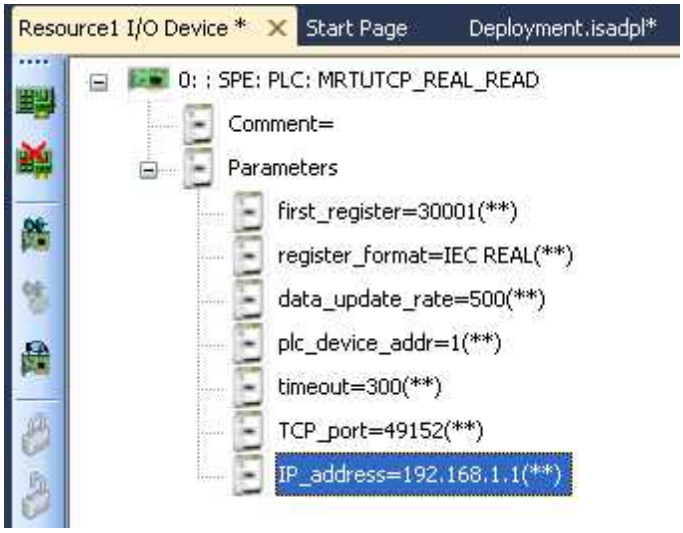
PLC Device I/O devices access Modbus PLC data in the following way:

- a PLC Device READ device reads PLC data from read-only and read-write registers
- a PLC Device WRITE device writes PLC data to read-write registers

I/O Devices are *strongly-typed*. Both DINT and REAL analog variables may **not** be mixed on the same Analog I/O device. However, conversion functions such as ANY_TO_DINT or ANY_TO_REAL may be used to perform a data type conversion in the user application as necessary.

7.5.2 Reading Modbus RTU in TCP Registers

Modbus RTU in TCP PLC Input device variables are updated at the start of an application scan. The value presented to the variables is the value returned by the PLC to the previous read request. This read may have occurred during previous application scans. The “data update rate” parameter on the I/O device sets the "scan" rate of the PLC data. The PLC communication status is updated if there is a status code returned from the PLC, or no response from the PLC after a data request by the SCADAPack E RTU. The status is cleared by the RTU upon successful communications. To catch transient status codes you can use code to store non-zero values.



Input Device Parameters

first_register*:* specifies the Modbus RTU in TCP PLC data registers to access when reading from PLC data into variables. The PLC data type accessed is the same as Modbus PLC Device I/O devices detailed in Section ***Serial Modbus Master I/O Devices*** 47.

register_format*:* specifies the Modbus RTU in TCP PLC data register type. The various types supported include IEC DISCRETE, 984 DISCRETE, IEC UINT, IEC INT, IEC DINT, IEC REAL and SWAP REAL. See the *SCADAPack E Target 5 Modbus Communication Interfaces* manual for more information.

data_update_rate*:* The units for this parameter are set in Milliseconds, and specify the rate at which the data for the Input device is extracted from the PLC. Individual I/O devices may have different data

update rates allowing prioritization of data extracted from a PLC device. The SCADAPack E RTU may not be able to read requested PLC data within the time set by the data update rate depending on the quantity of data to be read, rate of write requests and PLC communication speed. In this case the update rates will be slower.

plc_device_address: This parameter specifies the PLC device (unit) address. Modbus PLC devices accessed at the same IP address (e.g. via a Modbus bridge) needs to have a unique Unit address in order to be identified. Resources may access data from different units on the same IP address or at different IP addresses. In these cases a separate I/O device will be required for each device.

timeout: The Modbus RTU in TCP PLC device driver provides a parameter for specifying the communications timeout on an individual I/O device (i.e. the timeout applies to communications associated with that device). Where this value is "0", the PLC device driver will use the default timeout (1200 milliseconds). Units for this field are in milliseconds.

TCP_port: This parameter specifies the port number of the Modbus RTU in TCP server.

IP_address: This parameter specifies the IP network address that the SCADAPack E RTU connects to for communication with the PLC for this I/O device. Enter the IP address of the Modbus RTU in TCP PLC, or Modbus Bridge if applicable.

7.5.2.1 MRTUTCP_BOOL_READ I/O Device

Modbus RTU in TCP PLC 32 channel BOOL input device

Description

The MRTUTCP_BOOL_READ I/O device provides up to 32 IEC BOOL input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The digital channels can be connected to BOOL *VarInput* variables within an application. PLC data supported include coils, digital input status and holding registers. Each connected BOOL variable is updated continuously with the current value of the Modbus register. This information is cached internally by the RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td>
<table>
<tr><th>Enter Address</th><th>To ...</th></tr>
<tr><td>1 - 9999</td><td>Read Coils - Modbus Function Code 1</td></tr>
<tr><td>10001 - 19999</td><td>Read Status Register - Modbus Function Code 2</td></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table>
</td></tr>
<tr><td>plc_data_type</td><td>IEC DISCRETE, 984 DISCRETE</td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Port number of the Modbus RTU in TCP server (e.g. 49152)</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.5.2.2 MRTUTCP_DINT_READ I/O Device

Modbus RTU in TCP PLC 62 channel DINT input device

Description

The MRTUTCP_DINT_READ I/O device provides up to 62 IEC DINT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The analog input channels can be tied to DINT *VarInput* variables within an application. PLC data supported include inputs, and holding registers. Each connected DINT variable is updated continuously with the current value of two consecutive Modbus registers. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65534</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Port number of the Modbus RTU in TCP server (e.g. 49152)</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.5.2.3 MRTUTCP_INT_READ I/O Device

Modbus RTU in TCP PLC 124 channel INT input device

Description

The MRTUTCP_INT_READ I/O device provides up to 124 IEC INT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The analog input channels can be tied to INT *VarInput* variables within an application. PLC data supported include inputs, and holding registers. Each connected INT variable is updated continuously with the current value of the Modbus register. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Port number of the Modbus RTU in TCP server (e.g. 49152)</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.5.2.4 MRTUTCP_REAL_READ I/O Device

Modbus RTU in TCP PLC 62 channel REAL input device

Description

The MRTUTCP_REAL_READ I/O device provides up to 62 IEC REAL analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The analog input channels can be tied to REAL *VarInput* variables within an application. PLC data supported include inputs, and holding registers. Each connected REAL variable is updated continuously with the current value of two consecutive Modbus registers. This information is cached internally by the SCADAPack E RTU and made available to the application.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65534</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>register_format</td><td>IEC REAL, SWAP REAL</td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Port number of the Modbus RTU in TCP server (e.g. 49152)</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.5.2.5 MRTUTCP_UINT_READ I/O Device

Modbus RTU in TCP PLC 124 channel INT input device

Description

The MRTUTCP_UINT_READ I/O device provides up to 124 IEC UINT analog input channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via a TCP/IP connection. The analog input channels can be tied to UINT *VarInput* variables within an application. PLC data supported include inputs, and holding registers. Each connected UINT variable is updated continuously with the current value of the Modbus register. This information is cached internally by the SCADAPack E RTU and made available to the application.

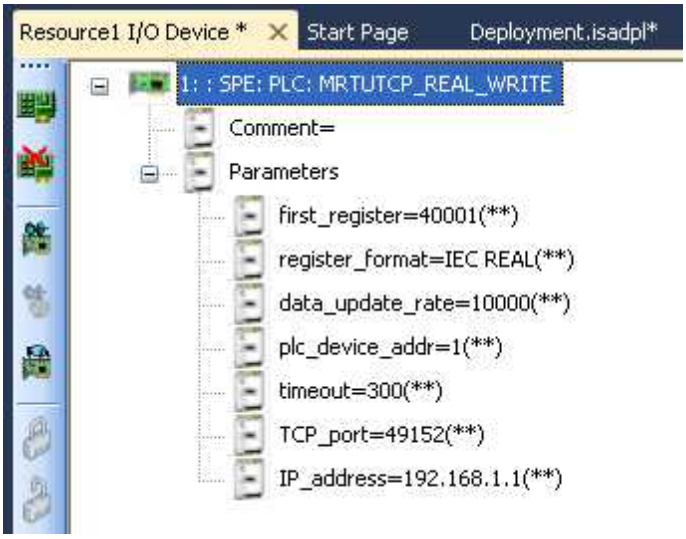
I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>30001 - 39999</td><td>Read Input Register - Modbus Function Code 4</td></tr>
<tr><td>40001 - 65535</td><td>Read Holding Register - Modbus Function Code 3</td></tr>
</table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Port number of the Modbus RTU in TCP server (e.g. 49152)</td></tr>
<tr><td>IP_address</td><td>IP address of the peripheral PLC device (e.g. 172.244.199.200)</td></tr>
</table>

7.5.3 Writing Modbus RTU in TCP Registers

Modbus RTU in TCP PLC Output device data is updated to the PLC when an application changes the value of a variable attached to the Output device. In addition, output device data is updated to the PLC under the following conditions:

- When the application starts output device data is written
- If the PLC does not respond to a control, it is re-sent until it is responded
- Output device data is rewritten at a background update rate



Output Device Parameters

first_register**:* specifies the Modbus RTU in TCP PLC data registers to access when reading from PLC data into variables. The PLC data type accessed is the same as Modbus PLC Device I/O devices detailed in Section ***Serial Modbus Master I/O Devices 47

***register_format**:* specifies the Modbus RTU in TCP PLC data register type.The various types supported include IEC DISCRETE, 984 DISCRETE, IEC UINT, IEC INT, IEC DINT, IEC REAL and SWAP REAL. See the *SCADAPack E Target 5 Modbus Communication Interfaces* manual for more information.

***plc_device_address**:* This parameter specifies the PLC device (unit) address. Modbus PLC devices accessed at the same IP address (e.g. via a Modbus bridge) needs to have a unique unit address in order to be identified. Resources may access data from different units on the same IP address or at different IP addresses. In these cases a separate I/O device will be required for each PLC device.

***data_update_rate**:* The unit for this parameter is set in Milliseconds, and specifies the rate at which the data for the Output device is written to the PLC. Between "*data_update_rate*" periods, data is only written to the PLC when the output variable values change. Individual I/O devices may have different data update rates allowing prioritization of data sent to PLC devices.

***timeout**:* The Modbus/TCP PLC device driver provides a parameter for specifying the communications timeout on an individual I/O device (i.e. the timeout applies to communications associated with that device). Where this value is "0", the PLC device driver will use the default timeout (1200 milliseconds). Units for this field are in milliseconds.

TCP_port: This parameter specifies the port number of the Modbus RTU in TCP server.

IP_address: This parameter specifies the IP network address that the RTU connects to for communication with the PLC for this I/O device. Enter the IP address of the Modbus/TCP PLC, or Modbus Gateway or Modbus Bridge, as applicable.

7.5.3.1 MRTUTCP_BOOL_WRITE I/O Device

Modbus RTU in TCP PLC 32 channel BOOL Output device

Description

The MRTUTCP_BOOL_WRITE I/O device provides up to 32 IEC BOOL output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The digital output channel can be tied to BOOL *VarOutput* variables within an application. PLC data supported include relays coils and holding registers. A target Modbus register is updated as necessary for each of the connected BOOL output variables.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table>
<tr><th>Enter Address</th><th>To...</th></tr>
<tr><td>1 - 9999</td><td>Write Coil - Modbus Function Code 5</td></tr>
<tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr>
</table></td></tr>
<tr><td>plc_data_type</td><td>IEC DISCRETE, 984 DISCRETE</td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_port</td><td>Port number of the Modbus RTU in TCP server (e.g. 49152)</td></tr>
<tr><td>IP_address</td><td>IP address of PLC device (e.g. 172.123.250.104)</td></tr>
</table>

7.5.3.2 MRTUTCP_DINT_WRITE I/O Device

Modbus RTU in TCP PLC 61 channel DINT output device

Description

The MRTUTCP_DINT_WRITE I/O device provides up to 61 IEC DINT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The analog output channels can be tied to DINT *VarOutput* variables within an application. PLC data supported include analog outputs and holding registers. Two consecutive Modbus registers are updated as necessary for each of the connected DINT output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65534</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
data_update_rate	units in milliseconds
plc_device_addr	Modbus Unit Identifier (1 - 254)
timeout	units in milliseconds
TCP_Port	Port number of the Modbus RTU in TCP server (e.g. 49152)
IP_address	IP address of PLC device (e.g. 172.123.250.104)

7.5.3.3 MRTUTCP_INT_WRITE I/O Device

Modbus RTU in TCP PLC 122 channel INT output device

Description

The MRTUTCP_INT_WRITE I/O device provides up to 122 IEC INT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The analog output channels can be tied to INT *VarOutput* variables within an application. PLC data supported include analog outputs and holding registers. A target Modbus register is updated as necessary for each of the connected INT output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
data_update_rate	units in milliseconds
plc_device_addr	Modbus Unit Identifier (1 - 254)
timeout	units in milliseconds
TCP_Port	Port number of the Modbus RTU in TCP server (e.g. 49152)
IP_address	IP address of PLC device (e.g. 172.123.250.104)

7.5.3.4 MRTUTCP_REAL_WRITE I/O Device

Modbus RTU in TCP PLC 61 channel REAL output device

Description

The MRTUTCP_REAL_WRITE I/O device provides up to 61 IEC REAL analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The analog output channels can be tied to REAL *VarOutput* variables within an application. PLC data supported include analog outputs and holding registers. Two consecutive Modbus registers are updated as necessary for each of the connected REAL output variables.

I/O Device Parameters:

first_register	<table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40,001 - 65534</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table>
register_format	IEC REAL, SWAP REAL
data_update_rate	units in milliseconds
plc_device_addr	Modbus Unit Identifier (1 - 254)
timeout	units in milliseconds
TCP_Port	Port number of the Modbus RTU in TCP server (e.g. 49152)
IP_address	IP address of PLC device (e.g. 172.123.250.104)

7.5.3.5 MRTUTCP_UINT_WRITE I/O Device

Modbus RTU in TCP PLC 122 channel UINT output device

Description

The MRTUTCP_UINT_WRITE I/O device provides up to 122 IEC UINT analog output channels for a SCADAPack E RTU to communicate with a Modbus PLC peripheral I/O device via TCP/IP. The analog output channels can be tied to UINT *VarOutput* variables within an application. PLC data supported include analog outputs and holding registers. A target Modbus register is updated as necessary for each of the connected UINT output variables.

I/O Device Parameters:

<table>
<tr><td>first_register</td><td><table><tr><th>Enter Address</th><th>To...</th></tr><tr><td>40001 - 65535</td><td>Write Holding Register - Modbus Function Code 16</td></tr></table></td></tr>
<tr><td>data_update_rate</td><td>units in milliseconds</td></tr>
<tr><td>plc_device_addr</td><td>Modbus Unit Identifier (1 - 254)</td></tr>
<tr><td>timeout</td><td>units in milliseconds</td></tr>
<tr><td>TCP_Port</td><td>Port number of the Modbus RTU in TCP server (e.g. 49152)</td></tr>
<tr><td>IP_address</td><td>IP address of PLC device (e.g. 172.123.250.104)</td></tr>
</table>

7.6 Allen Bradley DF1 PLC I/O Devices

The Allen-Bradley (Rockwell) PLCs communicate with the SCADAPack E RTU with DF1 protocol configured using **DF1_xxx** I/O devices through an RTU *PLC Device* serial port.

The DF1 registers are read and the return values cached in the RTU for access through an input device. Outputs are written from the RTU's output cache to the DF1 PLC. The SCADAPack E RTU's handling of the communications is the same as other PLC driver communications. The age and status of the data read from the DF1 PLC is present in RTU system points that can be accessed from within resources or external to the RTU.

A maximum of 200 PLC Device I/O devices (total of every PLC type) may be configured in total per Resource.

The DF1 Driver supports communications to the following Allen-Bradley PLC's:

- SLC 500 Series
- PLC 5 Series
- DF1 Generic PLC's

The **DF1_xxx** I/O devices use an RTU serial port configured as a *PLC Device* port to communicate with the Allen-Bradley.

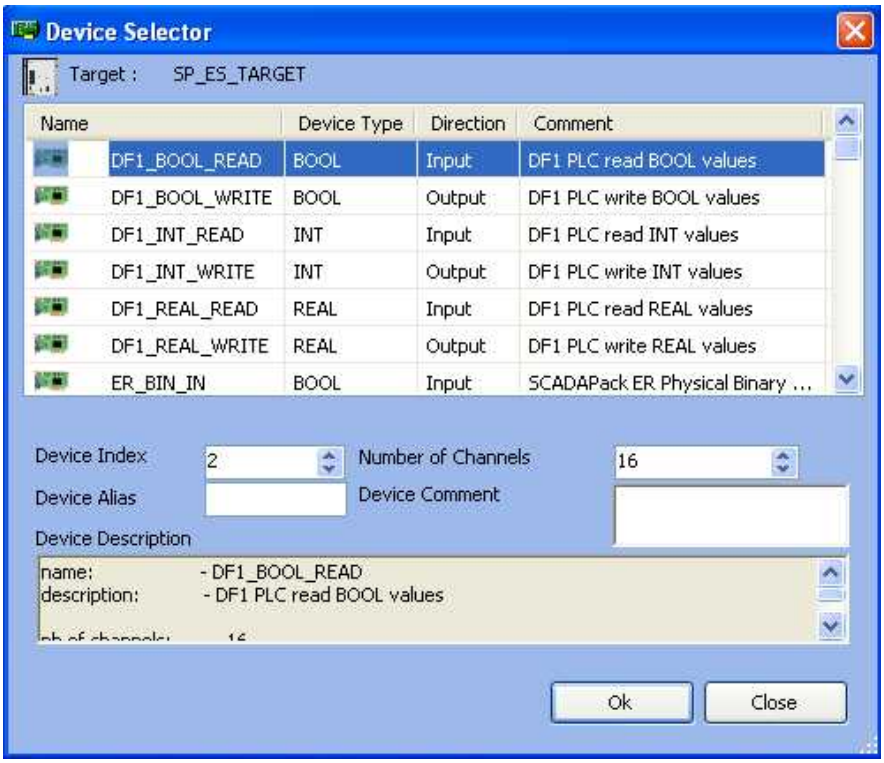
I/O Devices are *strongly-typed*. Both DINT and REAL analog variables may **not** be mixed on the same Analog I/O device. However, conversion functions such as ANY_TO_DINT or ANY_TO_REAL may be used to perform a data type conversion in the user application as necessary.

- ***Reading DF1 Registers*** 100
- ***Writing DF1 Registers*** 106

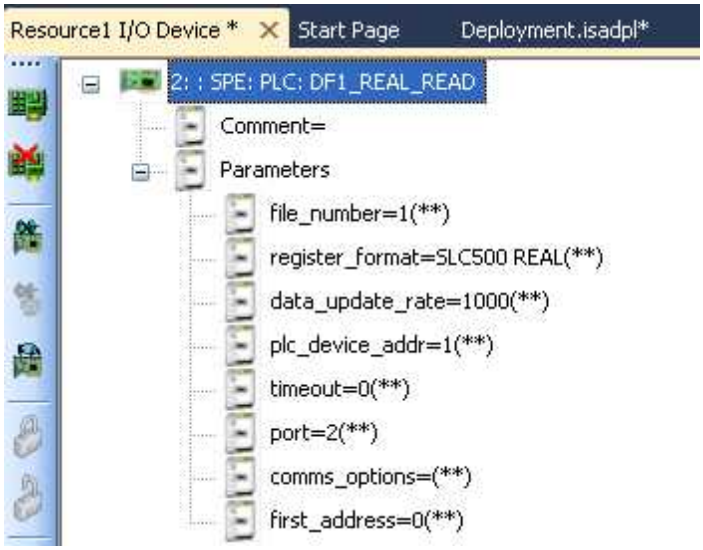
7.6.1 Reading DF1 Registers

The Input I/O Devices supported by the DF1 Driver are:

- 16 INT analog input
- 16 REAL analog input
- 16 BOOL input



These have the same basic layout as shown below.



The **file_number** field of the DF1 device (default 1) is the configurable file address of the required registers in the DF1 PLC.

The **register_format** field of the DF1 device (default SLC UINT for the AI devices, and SLC DISCRETE for the DI device) configures the device to communicate with the specified type of register in the specified PLC. Allowable values are outlined below:

Value	Description
SLC500 DISCRETE	Use on a digital device to communicate to a SLC500 PLC.
SLC500 INT	Use on an analog device to communicate to a SLC500 PLC. 16-bit signed value.
SLC500 REAL	Use on an analog device to communicate to a SLC500 PLC. 32-bit floating point value.
PLC5 DISCRETE	Use on a digital device to communicate to a PLC5 PLC.
PLC5 INT	Use on an analog device to communicate to a PLC5 PLC. 16-bit signed value.
PLC5 REAL	Use on an analog device to communicate to a PLC5 PLC. 32-bit floating point value.
GEN DISCRETE	Use on a digital device to communicate to a DF1 Generic PLC.
GEN INT	Use on an analog device to communicate to a DF1 Generic PLC. 16-bit signed value.

The **data_update_rate** field of the **DF1_xxx** I/O device (default 1000) is the configurable number of *milliseconds* after which the RTU will request element array values from the DF1 PLC. The RTU will also request data from the Allen-Bradley PLC constantly if the cache data age is greater than the **data_update_rate**. I.e. if communications are lost with the PLC, they are retried until the communications are restored.

The **plc_device_addr** (default 1) field of the device is the configurable address of the Allen-Bradley PLC.

The **timeout** field of the device driver provides a parameter for specifying the communications timeout on an individual I/O device (i.e. the timeout applies to communications associated with that device). Where this value is “0”, the PLC device driver will use the default timeout (1200 milliseconds). Units for this field are in milliseconds.

The **port** field of the device driver provides a parameter which defines which of multiple RTU “PLC Device” ports will be used to communicate with the PLC or peripheral device. If only one “PLC Device” port is configured, this field is ignored. PLC Device I/O devices not including this parameter can only be used when a single *PLC Device* port is configured on the SCADAPack E RTU.

The **comms_options** field is a string field that allows the user to set the local DF1 address, whether it’s half or full duplex, and whether it uses a CRC or BCC. The format for this string is as follows:

XXX YYYY ZZZ , where:

- XXX is the DF1 Address that the RTU will appear as (default is 0).
- YYYY is HALF or FULL for the duplex setting (default is FULL).
- ZZZ is CRC or BCC (default is CRC).

If any of the comms options fields are missing, then the default will be used for that parameter.

For Full Duplex operation set the DF1 address to be the address that you want the SCADAPack E RTU to appear a. However, for Half-Duplex operation set the DF1 address to be the 'Node Address' specified in the channel configuration of the PLC.

The **first_address** field of the device driver specifies the offset address of the device into the specified file.

7.6.1.1 DF1_BOOL_READ I/O Device

DF1 PLC 16 channel BOOL Input device

Description

The DF1_BOOL_READ I/O device provides up to 16 IEC BOOL input channels for a SCADAPack E RTU to communicate with an Allen Bradley PLC peripheral I/O device via a serial connection. The digital channels can be connected to BOOL *VarInput* variables within an application. PLC data supported include coils, digital input or holding registers. The BOOL variable is continuously updated with the Current State of the attached I/O. This information is cached internally by the RTU and made available to the application.

I/O Device Parameters:

file_number	DF1 File address
register_format	SLC500 DISCRETE, PLC5 DISCRETE, GEN DISCRETE
data_update_rate	Unit in milliseconds
plc_device_addr	DF1 Slave address (1-254)
timeout	Unit in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)
comms_options	XXX YYYY ZZZ where: XXX is the DF1 Address that the RTU will appear as (default is 0). YYYY is HALF or FULL for the duplex setting (default is FULL). ZZZ is CRC or BCC (default is CRC).
first_address	Offset in file for the first I/O device channel.

7.6.1.2 DF1_INT_READ I/O Device

DF1 PLC 16 channel INT Input device

Description

The DF1_INT_READ I/O device provides up to 16 IEC INT analog input channels for a SCADAPack E RTU to communicate with an Allen Bradley PLC peripheral I/O device via a serial connection. The analog channels can be connected to INT *VarInput* variables within an application. PLC data supported include analog inputs. The INT variable is continuously updated with the Current State of the attached I/O. This information is cached internally by the RTU and made available to the application.

I/O Device Parameters:

file_number	DF1 File address
register_format	SLC500 INT, PLC5 INT, GEN INT
data_update_rate	Unit in milliseconds
plc_device_addr	DF1 Slave address (1-254)
timeout	Unit in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)
comms_options	XXX YYYY ZZZ where: XXX is the DF1 Address that the RTU will appear as (default is 0). YYYY is HALF or FULL for the duplex setting (default is FULL). ZZZ is CRC or BCC (default is CRC).
first_address	Offset in file for the first I/O device channel.

7.6.1.3 DF1_REAL_READ I/O Device

DF1 PLC 16 channel REAL Input device

Description

The DF1_REAL_READ I/O device provides up to 16 IEC REAL analog input channels for a SCADAPack E RTU to communicate with an Allen Bradley PLC peripheral I/O device via a serial connection. The analog channels can be connected to REAL *VarInput* variables within an application. PLC data supported include analog inputs. The REAL variable is continuously updated with the Current State of the attached I/O. This information is cached internally by the RTU and made available to the application.

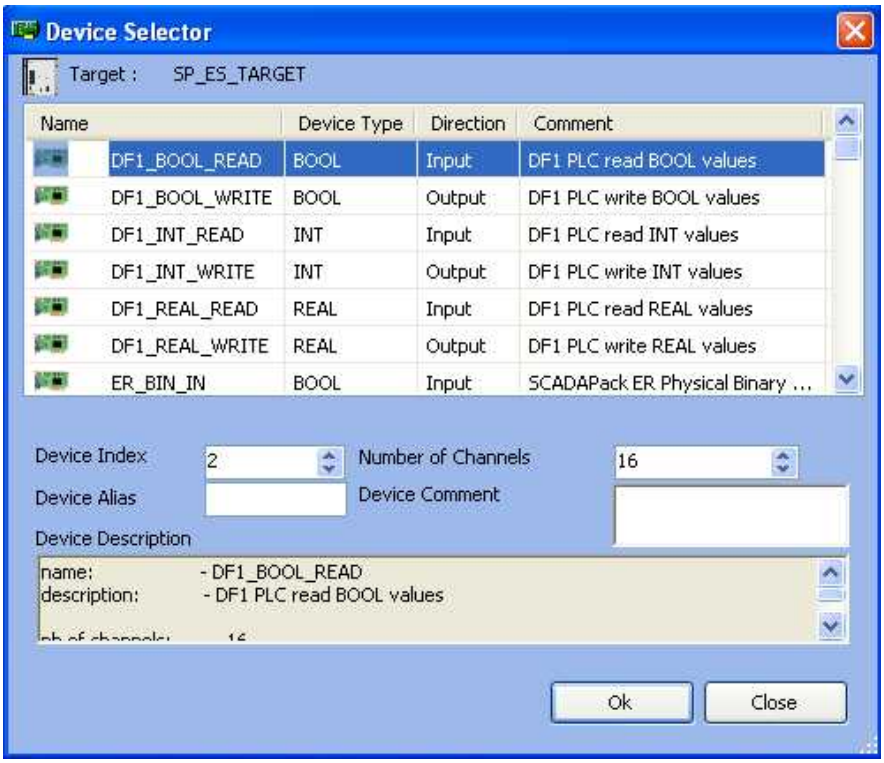
I/O Device Parameters:

file_number	DF1 File address
register_format	SLC500 REAL, PLC5 REAL
data_update_rate	Unit in milliseconds
plc_device_addr	DF1 Slave address (1-254)
timeout	Unit in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)
comms_options	XXX YYYY ZZZ where: XXX is the DF1 Address that the RTU will appear as (default is 0). YYYY is HALF or FULL for the duplex setting (default is FULL). ZZZ is CRC or BCC (default is CRC).
first_address	Offset in file for the first I/O device channel.

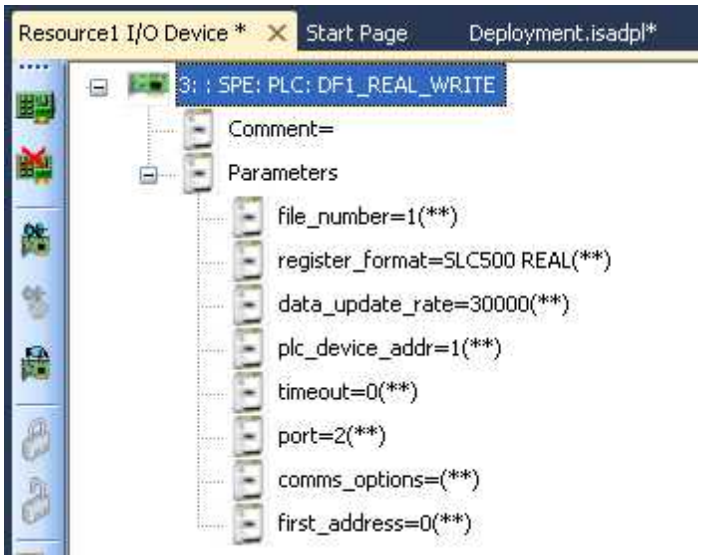
7.6.2 Writing DF1 Registers

The output devices supported by the DF1 Driver are:

- 16 BOOL output
- 16 INT analog output
- 16 REAL digital output



These devices have the same basic layout shown below.



These parameters are the same as described for the Input devices. The only difference is the **data_update_rate**. The unit for this parameter is in Milliseconds, and specifies the rate at which the data for the Output device is written to the PLC. Between "*data_update_rate*" periods, data is written to the PLC only when the output variable values change. Individual I/O devices may have different data update rates allowing prioritization of data sent to a PLC Device.

7.6.2.1 DF1_BOOL_WRITE I/O Device

DF1 PLC 16 channel BOOL Output device

Description

The DF1_BOOL_WRITE I/O device provides up to 16 IEC BOOL output channels for a SCADAPack E RTU to communicate with an Allen Bradley PLC peripheral I/O device via a serial connection. The digital channels can be connected to BOOL *VarOutput* variables within an application. PLC data supported include digital outputs. The *Current State* of the connected I/O is updated as necessary from the BOOL output variables.

I/O Device Parameters:

file_number	DF1 File address
register_format	SLC500 DISCRETE, PLC5 DISCRETE, GEN DISCRETE
data_update_rate	Unit in milliseconds
plc_device_addr	DF1 Slave address (1-254)
timeout	Unit in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)

comms_options	XXX YYYY ZZZ where: XXX is the DF1 Address that the RTU will appear as (default is 0). YYYY is HALF or FULL for the duplex setting (default is FULL). ZZZ is CRC or BCC (default is CRC).
first_address	Offset in file for the first I/O device channel.

7.6.2.2 DF1_INT_WRITE I/O Device

DF1 PLC 16 channel INT Output device

Description

The DF1_INT_WRITE I/O device provides up to 16 IEC INT analog output channels to a SCADAPack E RTU to communicate with an Allen Bradley PLC peripheral I/O device via a serial connection. The analog channels can be connected to an INT *VarOutput* variable within an application. The *Current State* of the connected I/O is updated as necessary from the INT output variables.

I/O Device Parameters:

file_number	DF1 File address
register_format	SLC500 INT, PLC5 INT, GEN INT
data_update_rate	Unit in milliseconds
plc_device_addr	DF1 Slave address (1-254)
timeout	Unit in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)
comms_options	XXX YYYY ZZZ where: XXX is the DF1 Address that the RTU will appear as (default is 0). YYYY is HALF or FULL for the duplex setting (default is FULL). ZZZ is CRC or BCC (default is CRC).
first_address	Offset in file for the first I/O device channel.

7.6.2.3 DF1_REAL_WRITE I/O Device

DF1 PLC 16 channel REAL Output device

Description

The DF1_REAL_WRITE I/O device provides up to 16 IEC REAL analog output channels to a SCADAPack E RTU to communicate with an Allen Bradley PLC peripheral I/O device via a serial connection. The analog channels can be connected to an REAL *VarOutput* variable within an application. The *Current State* of the connected I/O is updated as necessary from the REAL output variables.

I/O Device Parameters:

file_number	DF1 File address
register_format	SLC500 REAL, PLC5 REAL
data_update_rate	Unit in milliseconds
plc_device_addr	DF1 Slave address (1-254)
timeout	Unit in milliseconds
Port	0 = port0 (SCADAPack ES/SCADAPack ER only) 1 = port1 2 = port2 3 = port3 4 = port4 (SCADAPack ES/SCADAPack ER only)
comms_options	XXX YYYY ZZZ where: XXX is the DF1 Address that the RTU will appear as (default is 0). YYYY is HALF or FULL for the duplex setting (default is FULL). ZZZ is CRC or BCC (default is CRC).
first_address	Offset in file for the first I/O device channel.