Industrial Control

Modbus TCP Client for Ctek Automation Control

Tech Note – TN9042

November 9, 2020

Version 1.001

1891 N. Gaffey St. Ste. E
San Pedro, CA 90731

p. 310.241.2973

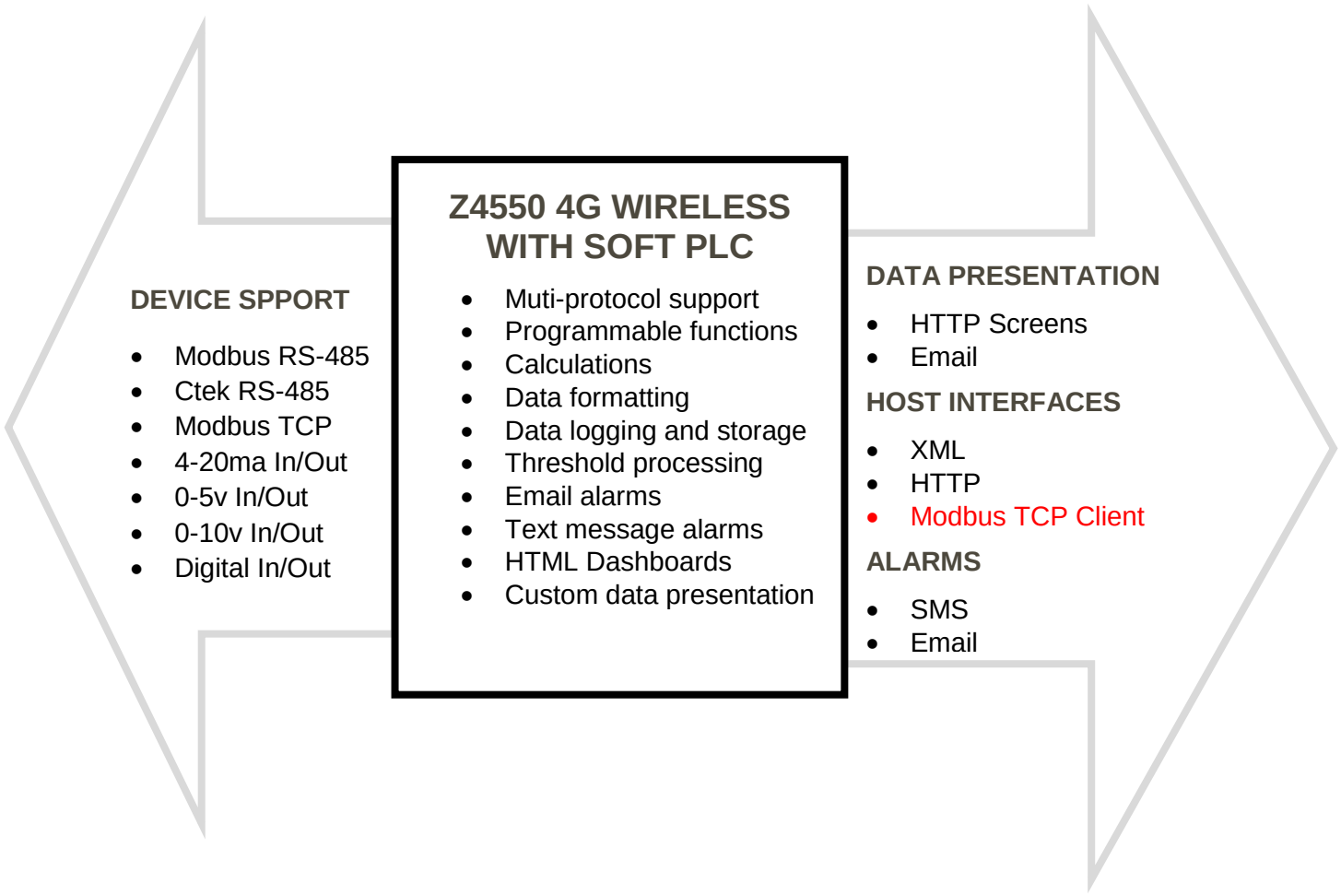
support@ctekproducts.com
www.ctekproducts.com

Table of Contents

Introduction

Ctek Z4550 cellular application platforms contain an Automation Control environment that allows anyone to create complete, autonomous, monitoring and control systems anywhere they may be required. Many of these monitoring and control systems act as complete, stand-alone SCADA systems and this has historically made it unnecessary to communicate with the typical SCADA server. However, in more complex environments, there are often servers in place that act as central monitoring and control points for numerous sub-systems. In support of this, Ctek has implemented a Modbus TCP client that allows the typical SCADA server to easily monitor and control functions that are hosted on Ctek's Automation Control.

Since Ctek's Automation Control environment acts as a complete SCADA system, it still performs all of its server functions while communicating downstream to another SCADA server via the Modbus TCP Client. This means that the Ctek Z4550 will continue to support its own network of devices over Modbus RS-485, Modbus TCP master and all of the other sensor interfaces that Ctek makes available. Through the Modbus TCP Client, a remote SCADA server can reach out to control or monitor data from any of the devices being supported on Ctek's Automation Control environment.



Internal Data Types

Within Ctek's Automation Control raw data is brought in from a wide range of devices and transformed into application specific data so that calculations and decision logic may be created. For instance a binary value from an A/D input might be transformed into a floating point value representing gallons of water or a binary value from a pulse counting input might be transformed into a floating point value representing flow rate in gallons per minute. The following internal data types are supported as both Inputs and Outputs:

Digital Pin

A single bit with a value of one or zero. Digital Outputs may be used to set the value of external physical devices or may be user defined Virtual Inputs / Outputs.

Numeric Pin

A 32 bit floating point value or a 32 bit signed or unsigned integer value. Numeric Pins may be configured to provide 0, 2 or 6 places of fractional accuracy. It is important to note that setting

fractional accuracy to 0, sets the Pin to integer mode. Numeric values may originate from a wide range of A/D, pulse or Modbus register devices since simple conversion facilities allow values from any of those physical sources to be normalized and converted to meaningful engineering values. Numeric Inputs may also be user defined Virtual Inputs.

Hex Pin

A 32 bit integer value that may be processed in either 32 or 16 bit mode. Hex values may come from a wide variety of physical register based devices such as Modbus devices. Hex values may also be user defined Virtual Inputs / Outputs.

Modbus Client Data Type Mapping

Each internal data type provided by Automation Control is accessible through the Modbus TCP Client Support. This section defines all available mapping options and mapping options are also provided at the end of this section in table format:

- **Reading a Digital Input:**
    - Any Digital Input (Virtual or Physical) may be read.
    - A Digital Input may be read using the ***Modbus Read Discrete Inputs*** function or ***Modbus Read Coils function***.
    - Multiple Input reads are supported.
- **Writing a Digital Input:**
    - Only Virtual Digital Inputs may be written. Physical Inputs cannot be written.
    - A value may be written to a Virtual Digital Input using the ***Modbus Write Single Coil*** function or ***Modbus Write Multiple Coils*** function.
    - Multiple Digital Input writes are supported.
- **Reading a Digital Output:**
    - Any Digital Output (Virtual or Physical) may be read.
    - A Digital Output may be read using the ***Modbus Read Coils*** function or ***Modbus Read Discrete Inputs*** function.
    - Multiple Digital Output reads are supported.

- **Writing a Digital Output:**
    - Any Digital Output (Virtual or Physical) may be written.
    - A Digital Output may be written using the ***Modbus Write Single Coil*** function, or ***Modbus Write Multiple Coils*** function.
    - Multiple Digital Output writes are supported.
- **Reading a Numeric Input:**
    - Any Numeric Input (Virtual or Physical) may be read.
    - If the input source is any of the following, a Double Register (32 bit) read is required and may be done using either a ***Modbus Read Input Registers*** function or a ***Modbus Read Holding Registers*** Function.:
        - Ctek Analog Input
        - Ctek Pulse Input
        - Virtual Numeric Input
        - 32 bit Modbus holding or input register
        - 16 bit Modbus holding or input register - Each 16 bit Modbus holding or input register will be mapped to a Ctek 32 bit Numeric Input.
    - Multiple 32 bit register reads are supported.
    - If a Numeric Input is configured with a fractional scale set to 0, a 32 bit integer value will be returned.
    - If a Numeric Input is configured with a fractional scale other than 0, a 32 bit floating point value will be returned.
    - 16 Bit Option – A single register (16 bit) read may be performed on a Ctek Numeric Input using the ***Modbus Read Input Registers*** function or a ***Modbus Read Holding Registers*** function to perform a single 16 bit integer read. Multiple register reads are not supported in 16 bit mode.
- **Writing a Numeric Input:**
    - Only Virtual Numeric Inputs may be written. Physical Inputs cannot be written.
    - A value may be written to a Virtual Numeric Input using a Double Register (32 bits) write. The ***Modbus Write Multiple Holding Register*** function may be used to write the 32 bit value to the selected Virtual Input.
    - Multiple 32 bit register writes are supported.
    - If the Virtual Numeric Input being written is configured with a fractional scale set to 0, the two words will be interpreted as an integer value.

- If the Virtual Numeric Input being written is configured with a fractional scale other than 0, the two words will be interpreted as a floating point value.

- 16 Bit Option - A single register (16 bit) ***Modbus Write Single Holding Register*** function may be used to write an integer value to a single Virtual Numeric Input.

- **Reading a Numeric Output:**

  - Any Numeric Output (Virtual or Physical) may be read.

  - If the output destination is any of the following, a Double Register (32 bit) read is required and may be done using a ***Modbus Read Input Registers*** function or a ***Modbus Read Holding Registers*** Function:
    - Virtual Output
    - 32 bit Modbus holding register
    - 16 bit Modbus holding register - Each 16 bit Modbus holding register will be mapped to a Ctek 32 bit Numeric Output.

  - Multiple register reads are supported in 32 bit mode.

  - If a Numeric Output is configured with a fractional scale set to 0, a 32 bit integer value will be returned.

  - If a Numeric Output is configured with a fractional scale other than 0, a 32 bit floating point value will be returned.

  - 16 Bit Option – A single register (16 bit) read may be performed on a Ctek Numeric Output using the ***Modbus Read Input Registers*** function or a ***Modbus Read Holding Registers*** function to perform a single 16 bit integer read. Multiple register reads are not supported in 16 bit mode.

- **Writing a Ctek Numeric Output:**

  - Any Numeric Output (Virtual or Physical) may be written.

  - If the output destination is any of the following a Double Register (32 bit) write is required and may be done using a ***Modbus Write Multiple Holding Register*** Function:
    - Virtual Output
    - 32 bit Modbus holding register
    - 16 bit Modbus holding register - Each 16 bit Modbus holding register will be mapped to a Ctek 32 bit Numeric Output.

  - Multiple register writes are supported in 32 bit mode.

- If the fractional scale of the Numeric Output is 0, a 2 word (32 bit) ***Modbus Write Multiple Holding Registers*** function will be interpreted as a 32 bit integer write to the selected output.

- If the fractional scale of the Numeric Output is not 0, a 2 word (32 bit) ***Modbus Write Multiple Holding Registers*** function will be interpreted as a floating point write to the selected output.

- 16 Bit Option - A single register (16 bit) ***Modbus Write Single Holding Register*** function may be used to write an integer value to a single Ctek Numeric Output.

- **Reading a Ctek Hex Input:**

  - Any Hex Input (Virtual or Physical) may be read.

  - If the input source is any of the following, a Double Register (32 bit) read is required and may be done using a ***Modbus Read Input Registers*** function or a ***Modbus Read Holding Registers*** Function.:
    - Virtual Hex Input
    - 32 bit Modbus holding or input register
    - 16 bit Modbus holding or input register - Each 16 bit Modbus holding or input register will be mapped to a Ctek 32 bit Hex Input.

  - A 32 bit integer value will be returned.

  - 16 Bit Option – A single register (16 bit) read may be performed on a Ctek Hex Input using the ***Modbus Read Input Registers*** function or a ***Modbus Read Holding Registers*** function to perform a single 16 bit integer read. Multiple register reads are not supported in 16 bit mode.

- **Writing a Ctek Hex Input:**

  - Only Virtual Hex Inputs may be written. Physical Inputs cannot be written.

  - A value may be written to a Virtual Hex Input using a Double Register (32 bits) write. The ***Modbus Write Multiple Holding Register*** function may be used to write the 32 bit value to the selected Virtual Input.

  - Multiple 32 bit register writes are supported.

  - The 32 bit value will be interpreted as an integer.

  - 16 Bit Option - A single register (16 bit) ***Modbus Write Single Holding Register*** function may be used to write an integer value to a single Virtual Hex Input.

- **Reading Hex Output:**

  - Any Hex Output (Virtual or Physical) may be read.

- If the output destination is any of the following, a Double Register (32 bit) read is required and may be done using a ***Modbus Read Input Registers*** function or a ***Modbus Read Holding Registers*** Function:
  - Virtual Output
  - 32 bit Modbus holding register
  - 16 bit Modbus holding register - Each 16 bit Modbus holding register will be mapped to a Ctek 32 bit Hex Output.

- Multiple register reads are supported in 32 bit mode.

- A 32 bit integer value will be returned.

- 16 Bit Option – A single register (16 bit) read may be performed on a Ctek Hex Output using the ***Modbus Read Input Registers*** function or a ***Modbus Read Holding Registers*** function to perform a single 16 bit integer read. Multiple register reads are not supported in 16 bit mode.

- **Writing a Ctek Hex Output:**

  - Any Hex Output (Virtual or Physical) may be written.

  - If the output destination is any of the following a Double Register (32 bit) write is required and may be done using a ***Modbus Write Multiple Holding Register*** Function:
    - Virtual Output
    - 32 bit Modbus holding register
    - 16 bit Modbus holding register - Each 16 bit Modbus holding register will be mapped to a Ctek 32 bit Numeric Output.

  - Multiple register writes are supported in 32 bit mode.

  - The value written will be interpreted as a 32 bit integer write to the selected output.

  - 16 Bit Option - A single register (16 bit) ***Modbus Write Single Holding Register*** function may be used to write an integer value to a single Ctek Hex Output.

Physical Device to Modbus Mapping Summary

The following table presents a mapping of all physical devices running on Ctek's Automation Control.

Raw Data Source	Ctek Type	Modbus Data Type	Modbus Function
Ctek Digital Input, 1 bit	Digital	Discrete Input	Fn 02 – Read Discrete Inputs
Ctek Digital Output, 1 bit	Digital	Coil	Fn 01 – Read Coils Fn 15 – Write Multiple Coils Fn 05 – Write Single Coil
Ctek Analog Input	Numeric	Input Register Integer or * Input Register Floating point	Fn 04 – Read Input Registers
Ctek Pulse Input	Numeric	Input Register Integer or * Input Register Floating point	Fn 04 – Read Input Registers
Modbus 16 bit Input	Numeric	Input Register Integer (16 bit)	Fn 04 – Read Input Registers
Modbus 16 bit Holding	Numeric	Holding Register Integer (16 bit)	Fn 03 – Read Holding Registers Fn 03 – Read Holding Registers Fn 16 – Write Multiple Registers Fn 06 – Write Single Register
Modbus 32 bit Input	Numeric	Read Input Register Integer (32 bit) or *Read Input Register Floating point	Fn 04 – Read Input Registers
Modbus 32 bit Holding	Numeric	Holding Register Integer (32 bit) or Holding Register Floating point	Fn 03 – Read Holding Registers Fn 16 – Write Multiple Registers
Modbus 16 bit Input	Hex	Input Register Integer (16 bit)	Fn 04 – Read Input Registers
Modbus 16 bit Holding	Hex	Holding Register Integer (16 bit)	Fn 03 – Read Holding Registers Fn 16 – Write Multiple Registers Fn 06 – Write Single Register
Modbus 32 bit Input	Hex	Input Register Integer (32 bit)	Fn 04 – Read Input Registers
Modbus 32 bit Holding	Hex	Holding Register Integer (32 bit)	Fn 03 – Read Holding Registers Fn 16 – Write Multiple Registers

*In automation control, floating point values must have a decimal scale defined of "2" or "6". A decimal scale of "0" will result in the numeric value being treated as an integer.

Virtual Device to Modbus Mapping Summary

Ctek Automation Control makes extensive use of Virtual I/O. Virtual I/O can be thought of as variables or computational blocks that can be tied together through simple read or write functions. In some cases, a Modbus host may be writing to or reading from a Virtual I/O that is performing some sort of conversion or processing of data from a physical device. The following table presents a mapping of all Virtual I/O available on Ctek's Automation Control

Internal Virtual Device	Supported Modbus Data Types	Modbus Function
Virtual Digital Input or Output	Discrete Input	Fn 02 – Read Inputs
	Coil	Fn 01 - Read Coils Fn 05 – Write Single Coil Fn 15 – Write Multiple Coils
Virtual Numeric Input or Output	Input Register Integer (32 bit or 16bit), Floating point	Fn 04 – Read Input Registers
	Holding Register Integer (32 bit or 16bit), Floating point	Fn 03 – Read Holding Registers Fn 16 – Write Multiple Holding Fn 06 – Write Single Holding
Virtual Hex Input or Output	Input Register Integer (32 bit or 16 bit)	Fn 04 – Read Input Registers
	Holding Register Integer (32 bit or 16 bit)	Fn 03 – Read Holding Registers Fn 16 – Write Multiple Holding Fn 06 – Write Single Holding

*In automation control, floating point values must have a decimal scale defined of "2" or "6". A decimal scale of "0" will result in the numeric value being treated as an integer.

Register Addressing

Within Ctek's Automation Control, both input addresses and output addresses start at 1. In order to operate within the standard I/O functions provided by Modbus, it is necessary to separate input and outputs into their own address range.

The default configuration of Ctek's Modbus TCP Client leaves the base address of inputs starting at 1 but moves the base address of outputs to 1000. This parameter may be adjusted if necessary. This parameter and other addressing configuration parameters will be defined in a following section on configuration parameters.

Since Ctek Numeric and Hex data elements are 32 bit, the address must be multiplied by 2 to comply with standard Modbus addressing. Digital data elements will have the same

- **Input example** - if you wish to address a Numeric or Hex Input configured at Ctek address 2, the Modbus address would be 4 (address * 2). You would use address 4 to retrieve the 32 bit value. You would also use address 4 if you were to do a single register (16 bit) read. In that case you will be retrieving the least significant portion of the 32 bit value.
- **Output example** - if you wish to address a Numeric or Hex Output configured at Ctek address 2, the Modbus address would be 1004 (address * 2 + 1000. You would use address 1004 to retrieve or write the 32 bit value. You would also use address 1004 if you were to do a single register (16 bit) read or write. In that case you will be retrieving or writing the least significant portion of the 32 bit value.

I/O Type	Input Address Range	Output Address Range
Holding Coil	00001 - 00999	01001 - 01999
Discrete Input	10001 - 10999	11001 - 11999
Input Register	30001 - 30999	31001 - 31999
Holding Register	40001 - 40999	41001 - 41999

IP Address and Port

Modbus TCP Client is configured to listen for incoming connections on both the Ethernet LAN and the wireless WWAN. The default listen port is set to 1502 but the port number may be changed if necessary. If Ctek Automation Control is acting as both a Modbus TCP Master and Modbus TCP Client, be careful to avoid port conflicts.

Configuration Parameters

There are a number of parameters associated with Modbus TCP Client that may need to be configured for certain applications. These parameters are kept in an internal file that can be modified if necessary. In order to modify them, you will use the TCO editor function which is documented in Ctek Application Note APN001 – TCOPlus Features.

The configuration file is /etc/conf.d/opt.rtu_modbus_tcp_client and contains the following parameters:

RESPTOSEC and RESPTOUSEC

These parameters define response timeout second and response timeout microseconds. This defines how long the Modbus Client will wait for a response before timing out.

BYTEOSEC and BYTEOUSEC

These parameters define the maximum time allowed between received characters.

ALWAYSON

These parameters define the maximum time allowed between received characters.

PORT

This parameter sets the TCP listen port for Modbus Client.

DEBUG

If Debug is set to 1 (on), many diagnostic messages will be generated in syslog. The logread command may be used to review them. Debug should always be set to 0 (off) during normal operation.

COFF, BOFF, HOFF, IOFF

Coil offset, Bit offset, Holding offset, and Input offset allow you to assign an offset to each class of Modbus device type when necessary. As an example, if a server requires that holding registers have a base address of 400001 and Ctek Automation has holding registers located at location 1001, you can set HOFF=”399000”

WOFF

As mentioned earlier, both input and output addresses start at 1 in Automation Control. WOFF adjusts the offset of outputs based on its value. By default, WOFF is set to 1000 giving the first output an address of 1001.

WSWAP

Defaults to “0” (no swapping). This parameter controls the read order of 32 bit integers. When set to “1” (swap 32 bit read order), the second register is returned as the first. Some configurations may see incorrect values for 32 bit integers. Changing this value to “1” may resolve the issue.

Configuring with other SCADA software

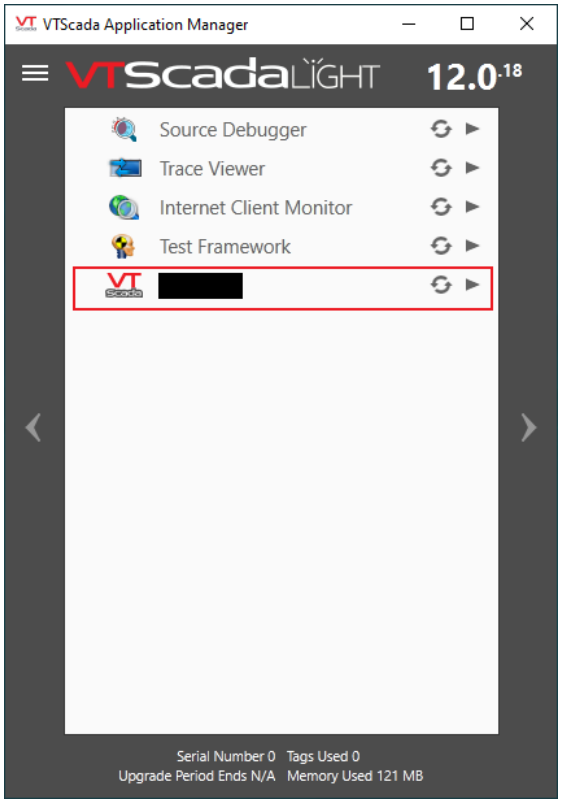
VT SCADA

SCADA software is a common use for the Modbus TCP Client. Instructions for an example application created with VT Scada are provided below. More information for creating VT Scada applications is available at VTScada.com/help

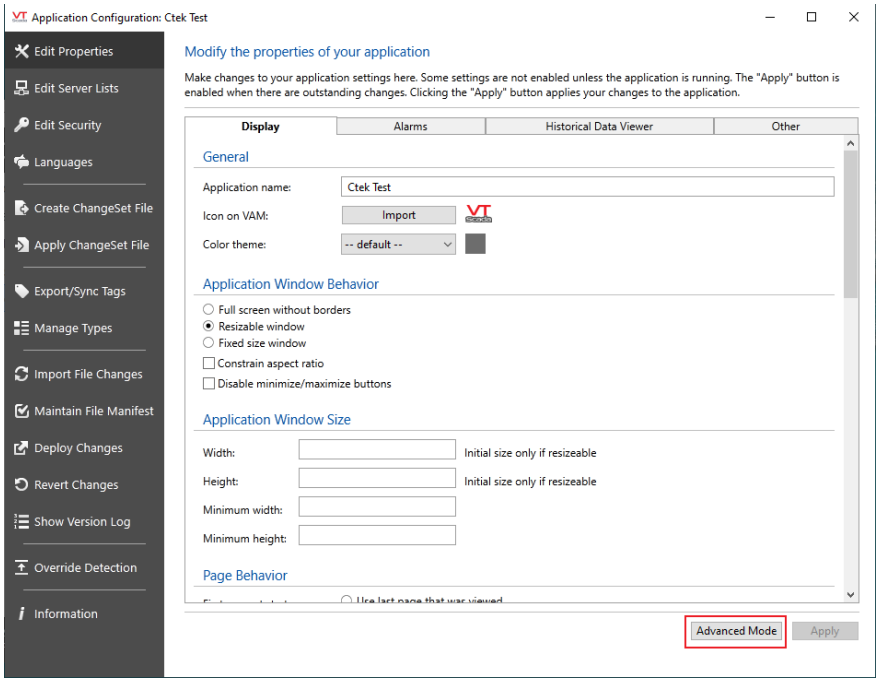
VT Scada reads several addresses at once through a strategy called coalescing. Coalescing may cause issues in situations where a 16 bit integer follows either a floating point value or a 32 bit integer. To solve these issues, several changes must be made to the VT Scada configuration to prevent issues when reading and writing.

Change max block read length:

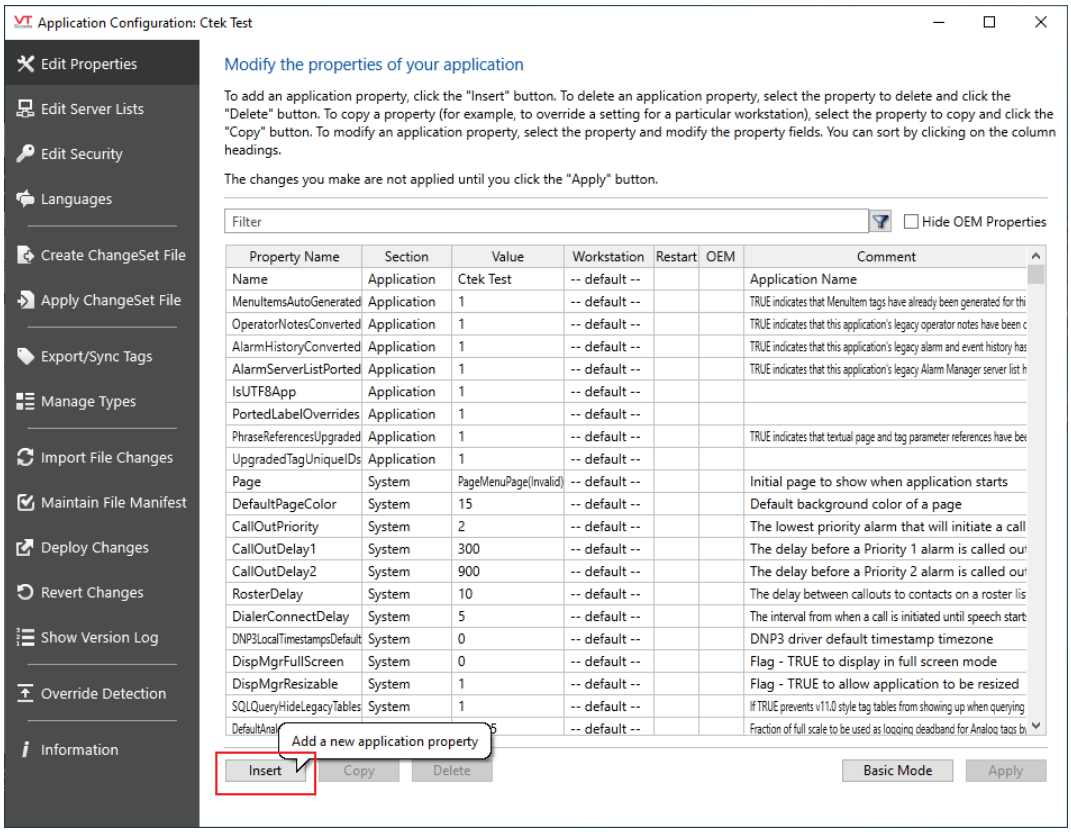
Right Click your application in the VT Scada application viewer menu.



Select the edit properties menu. At the bottom right of your screen, click the “Advanced Mode” button.



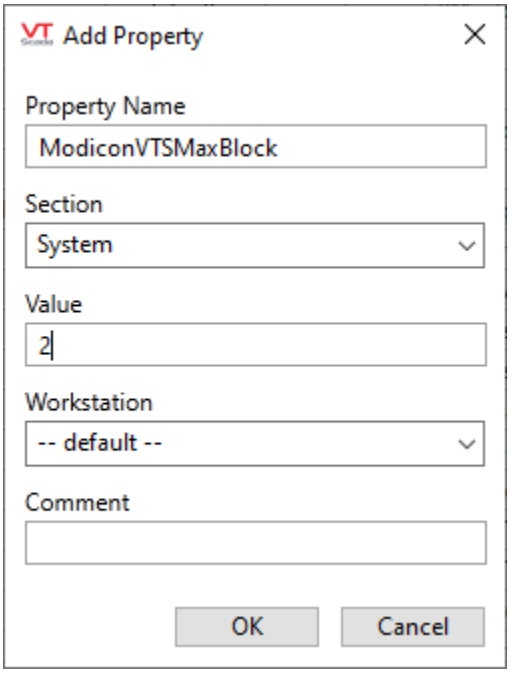
Then, select the insert button at the bottom left.



On the resulting popup, type in the following information:

Property Name: ModiconVTSMaxBlock

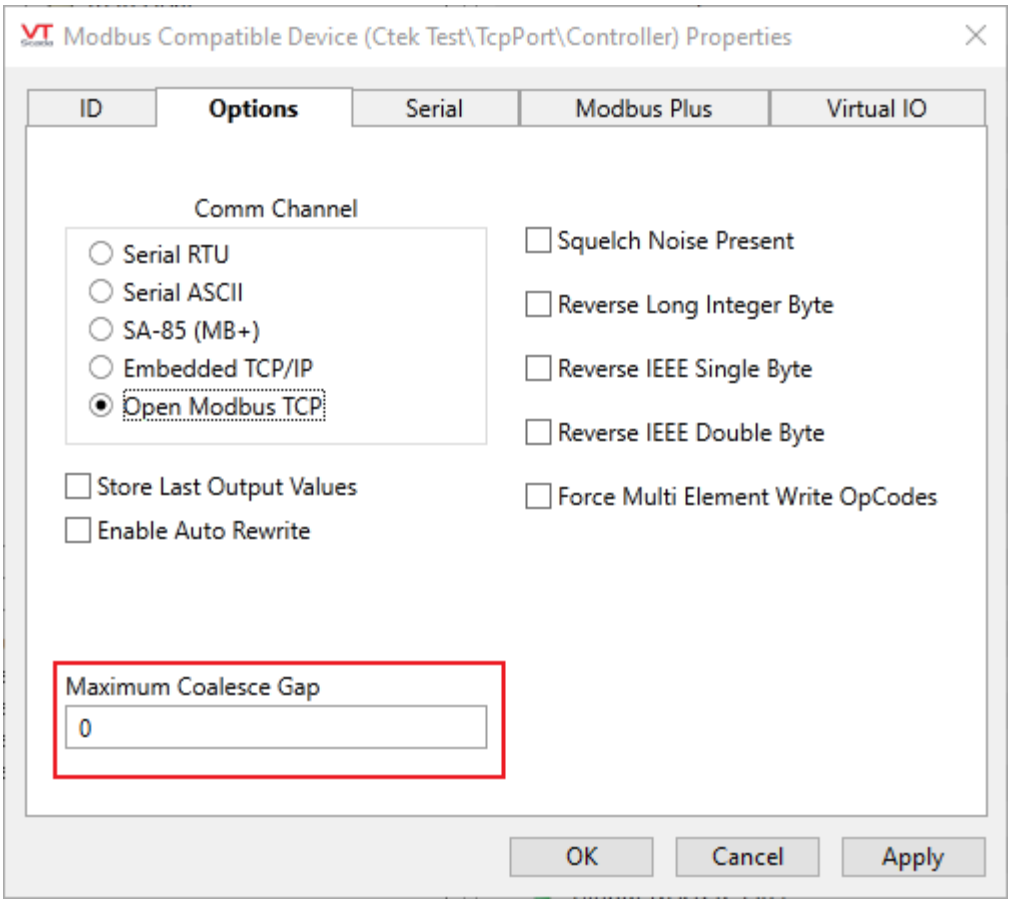
Value: 2



Finally Click Ok > Apply > Ok. You may now run your application in VT Scada.

Change the Max Coalesce Gap

Ctek Controllers do not support reading registers that are not explicitly declared. To fix this, the max coalesce gap must be set to “0”. When setting your Modbus Device Controller Driver, Select the IO tab and change the Max Coalesce Gap to a value of 0.



Make sure all addresses point to the correct register

1. When adding I/O tags in VT Scada, make sure all read / write addresses are pointing to the earlier of the two defined registers for the pin you are trying to access. The easiest way to calculate the correct register number is by taking the pin number as it appears on your Ctek Sky Router, multiply it by 2, and then add 1.

VT Scada Register Address = (Pin Number * 2) + Modbus 5 digit address range + 1

EXAMPLES

Ctek Pin Type	Example Pin Number	Corresponding Register Address
Digital	001	00003
Numeric	013	40027 (Holding Register)
Hex	007	30015 (Input Register)

Using data suffixes to read Ctek internal types:

2. VT Scada uses address suffixes instead of calling Modbus functions directly. The following table will show how to read Ctek internal types using the VT Scada suffixes.

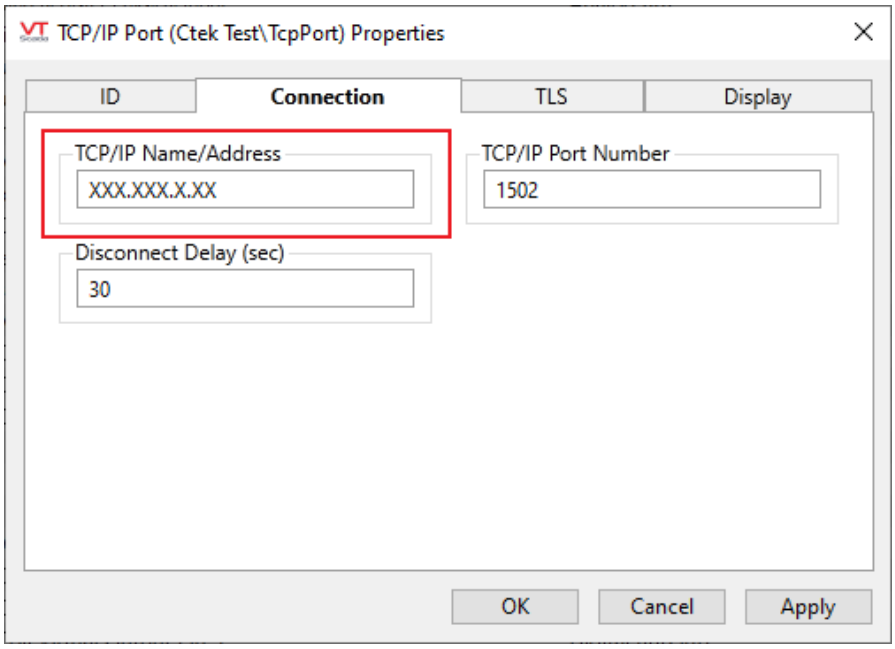
Ctek Internal Type	Available Suffixes
Digital	No suffix is needed for digital pins.
Numeric 16 Bit Integer	/UWORD (used for unsigned integers) /SWORD (used for signed integers)
Numeric 32 Bit Integer	/UDWORD (used for unsigned integers) /SDWORD (used for signed integers)
Numeric Floating Point	/Float
Hex	/UDWORD

Example

An example application for VT Scada along with an example Automation Control Configuration is available for your convenience. . You can request the example by contacting Ctek Inc. via phone at (310) 241-2973 or by the Ctek support link at www.ctekproducts.com/contact.

The VT Scada application provided will work with the Ctek Configuration provided. To install the VT Scada application, unzip the provided example to any folder on your computer. Add a new application in the VT Scada menu. When adding a new application, select the “Advanced” option and click “Next”. Find the example that you unzipped and select it. Then select “Next” and “Finish”. The application should now be available in the VT Scada Menu.

The only change that needs to be made to the application is changing the properties of the “TcpPort” tag to point to an IP address with the Ctek Controller.



The Automation Control Configuration contains a module for physical devices. This configuration is designed with the ICP DAS tm-DA1P1R1 interface module as the physical device. The physical device is assigned to the 4th module with a numeric input / output on pin 25 and a digital input / output on pin 29. By default this module and the physical tags within VT Scada are disabled. They are disabled because the tm-DA1P1R1 is not mandatory to test the configuration, various hardware may be used. Different hardware will require different steps to properly enable. More information for enabling the physical I/O Module and for attaching external hardware can be found in Tech Notes relating to configuring Automation Control.