U-CON Profile Development

ASCII Commands for interfacing host PCs (i.e. HMIs) to Keyence CV3002 controllers

U-CON Profile Development
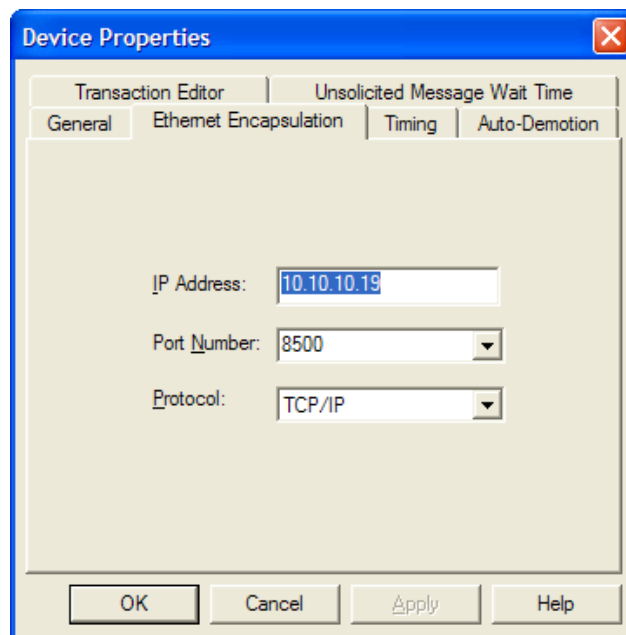# Keyence CV3002 controller

Overview

The purpose of this U-CON profile is to provide an interface from HMIs and other OPC client enabled applications to Keyence CV3002 controllers.

Communications

Communications Parameters

This device provides both an RS232 interface and an Ethernet port. The profile was tested with the Ethernet connection but works via the RS232 connection provided that the RS232 port is enabled in the controller, the proper cable wiring exists between controller and PC, and finally that the user disables Ethernet Encapsulation in the Channel properties of the U-CON profile (then set serial communication settings to match settings in the controller).

To setup the demo controller, we connected the Remote Control Console and a monitor to the controller, then referenced the CV3002 User's Manual for instructions on the Ethernet Interface (i.e. IP and subnet mask settings). In the sample U-CON project, edit the IP and Port fields below to match your Ethernet enabled controller.

Communications Data Speed

Ethernet between PC and controller is obviously very fast. However, please review the CV3002 User's Manual provided by Keyence to understand which commands could affect the measurement process time of your controller.

Communications Procedures

1. All processing in this protocol is *Solicited* meaning the host application sends commands to the controller (via U-CON) to change or request data.

2. Due to the nature of high speed controllers for vision applications, it is not recommended to constantly poll the controller for data. Therefore, the profile was designed such that the user must write a value (i.e. 1) to "Send[r/w]Command" tags which will trigger the U-CON profile to send the command to the controller.

   Note: The only exception to this (in this U-CON profile) is the M0 command which is sent to the controller at 1 second intervals to request Output Measurement result data (same data available from most recent Trigger). Refer to the Group called "Meas_Output_M0_polled" to review more closely.

3. Many commands require user input such as Program or Window numbers (000-127), Camera number (1-4), or Memory Type (IN or CF) prior to triggering the related "SendxyzCommand" tag.

4. U-CON provides Scratch Buffers to allow you to exchange information between transactions defined for that device. This profile uses scratch buffers extensively to provide User Input (as mentioned in note 3 above) and to provide a way to handle/display ASCII 2-digit Response and **Error Codes** sent from the controller. Refer to the U-CON project properties (File | Project Properties) for a list of the Scratch Buffers used.

5. At the time of developing this profile, U-CON did not contain a way to receive Image Data so the following commands are not supported:

   - Trigger commands with Auto Image Output set: T1 & T2

   - Image Data output via comm ports: BC, CM (note: CF is supported)

   - Reading Image Data: BR

   - Saving/Loading Program Data: SW & SR

   - Saving/Loading Global Settings Data: SB & SA

Packet Configuration

The following section will describe how a sample command is configured and what the response is that comes back. All supported commands use a Carriage Return (CR) to terminate the packet. Controller response commands use the CR to terminate packets as well, so the U-CON profile uses them in Read Response commands as shown below.
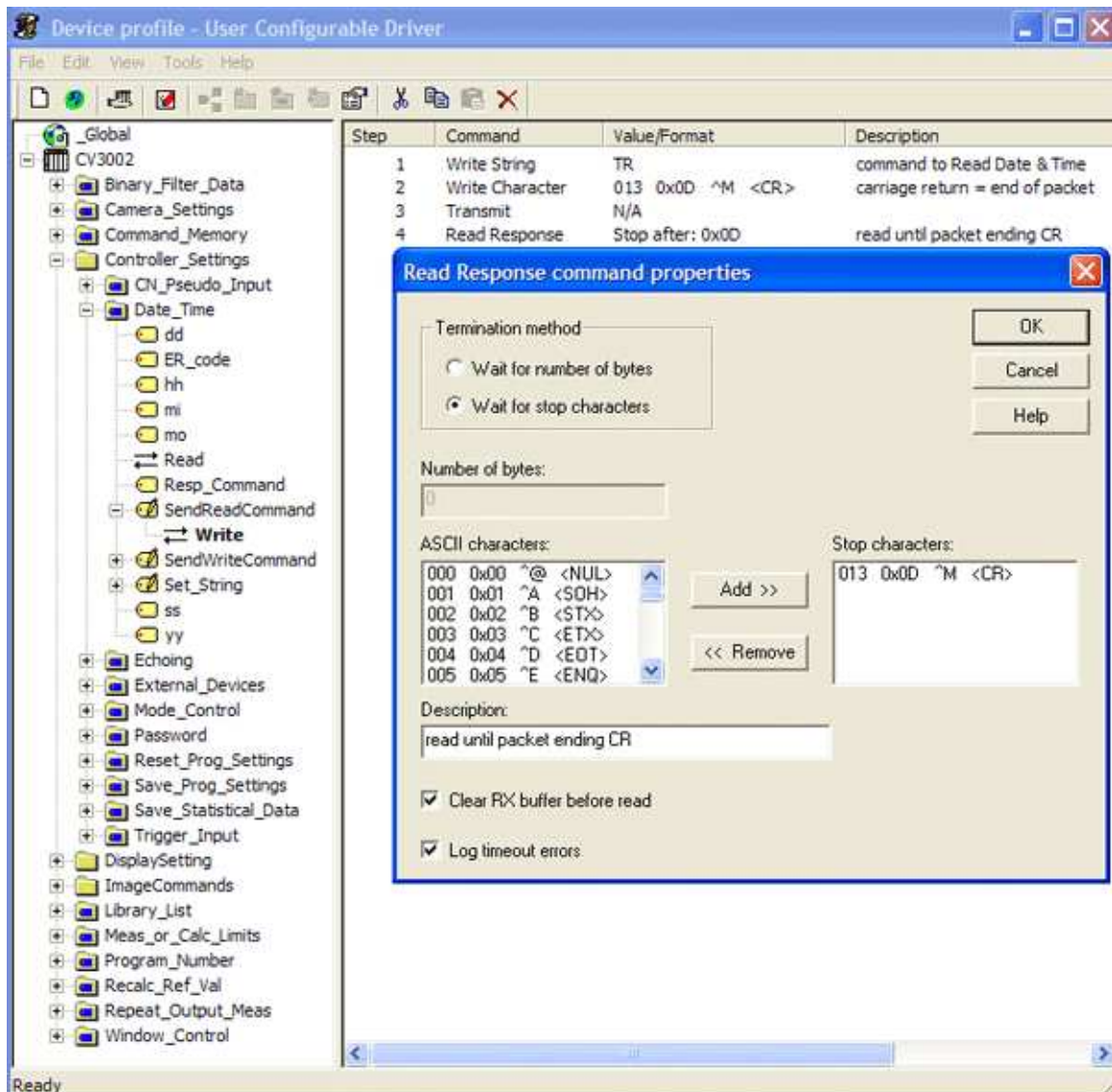
Reading date and time - TR

This command reads the date and time currently set in the controller.

....

Send: TR CR

Receive: TR , yy , mo , dd , hh , mi , ss CR

Please note: Spaces in the Send / Receive samples above are shown for clarity. The actual commands have no spaces so the Receive packet above would be 21 bytes long.
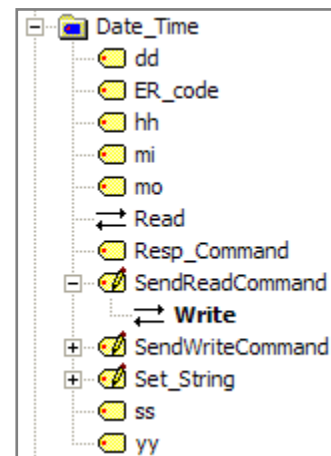
Here is a **sample of the Hex byte stream** captured in the Channel Diagnostics window of the U-CON server application for the Send / Receive sample above.

....

TX: 54 52 0D

RX: 54 52 2C 30 38 2C 30 36 2C 31 31 2C 31 36 2C 33 30 2C 31 32 0D

The picture to the right shows where the Write transaction for the tag "SendReadCommand" can be found in the "Date_Time" Tag Block.

....

Below is what the transaction steps look like from the U-CON Transaction Editor.

Step	Command	Value/Format	Description
1	Write String	TR	command to Read Date & Time
2	Write Character	013 0x0D ^M <CR>	carriage return = end of packet
3	Transmit	N/A	
4	Read Response	Stop after: 0x0D	read until packet ending CR
5	Copy Buffer	1 - end of data	copy RX: packet to scratch buffer 2

# The Profile

Profile / Transaction Design

This profile was designed with Error Handling tags associated with each Tag / Command group. It could be redesigned to use fewer Scratch Buffers with Error Handling shared across all Tags / Commands. Users who are concerned with high tag counts in their HMI application may want to consider the shared Error Handling design.

Earlier a sample of the SendReadCommand for Date_Time was shown. Below is the **Read** transaction to handle the response from the controller.

Step	Command	Value/Format	Description
1	Update Tag	Set_String	update with 17digit format from scratch buffer3
2	Update Tag	SendWriteCommand	update with "write 2 send" from scratch 254
3	Update Tag	SendReadCommand	update with "write 2 send" from scratch 254
4	Update Tag	Resp_Command	update from scratch 2
5	Comment	********************	
6	Test String	TR	Continue else Go To Test_for_TW
7	Update Tag	yy	byte 4
8	Update Tag	mo	byte 7
9	Update Tag	dd	byte 10
10	Update Tag	hh	byte 13
11	Update Tag	mi	byte 16
12	Update Tag	ss	byte 19
13	Update Tag	ER_code	update with "ok" from scratch buffer 255
14	End	N/A	
15	Comment	********************	
16	Label	Test_for_TW	
17	Test String	TW	Continue else Go To Test_for_ER
18	Update Tag	ER_code	update with "ok" from scratch buffer 255
19	End	N/A	
20	Comment	********************	
21	Label	Test_for_ER	
22	Test String	ER	Continue or End
23	Update Tag	ER_code	display nn: ASCII 2-digit error code (byte 7 & 8)
24	End	N/A	

How to Use the Device Profile

We recommend that users first use the free OPC Quick Client (launch from the toolbar of KEPServerEX or U-CON Protocol Server) to prove successful communications with the controller.
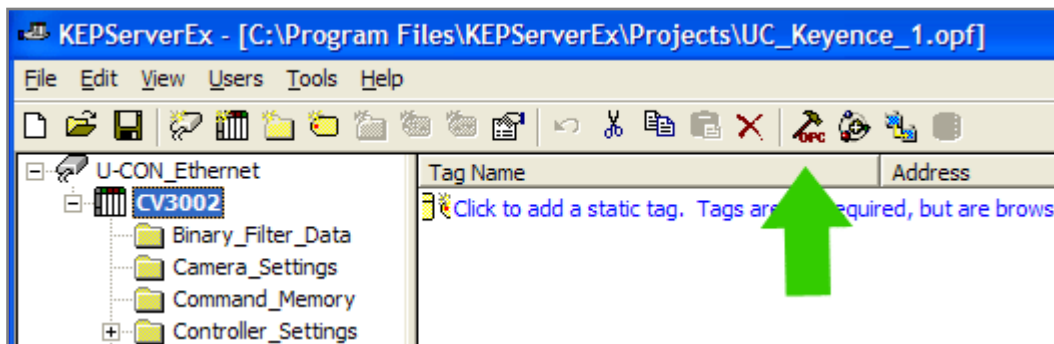
Here is the tag view in the Quick Client. To trigger the SendReadCommand right click over the tag after selected, then select Synchronous or Asynchronous Write. Next enter a value to trigger the write (any character will trigger the write command shown on page 4 of this document).

After connectivity is proven and the basic method of sending commands to the controller is understood, then connect your HMI or client application to the U-CON application. Sample Connectivity Guides (.pdfs) can be downloaded from the Support section of Kepware's website. **Reminder:** Review your CV3002 User's Manual to fully understand what commands make sense for your application. Be careful in using Mode change commands and Enable / Disable commands with controllers in production use (user's responsibility to prevent data loss or unwanted program changes).