ID: QA32494 | Access Levels: Everyone

PanelView™ 800 to a ControlLogix® 1756-L85E processor over Modbus TCP

Document ID	QA32494
Published Date	03/04/2020

Summary

PanelView™ 800 to a ControlLogix® 1756-L85E processor over Modbus TCP

Question

Is there a way to have a PanelView™ 800 communicate with a ControlLogix® 1756-L85E?

Environment

- PanelView™ 800 2711R-T7T/A (Firmware 4.014)
- ControlLogix® 1756-L85E/B (Firmware 30.011)
- Connected Components Workbench™ v10.01
- Studio 5000® v30.00.00

Answer

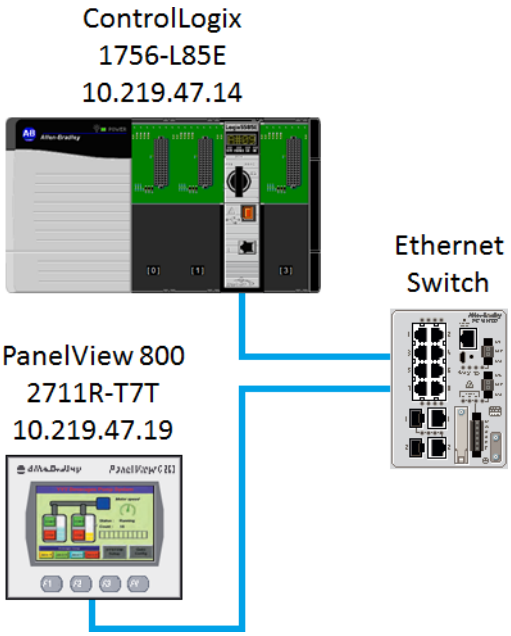
The attached examples, and following information provide an overview on how to tackle the problem.

For this example, it is assumed that the user is familiar with Connected Components Workbench, Studio 5000, and the general concepts of how a Modbus TCP network works. So the example will not go into deep detail on programming steps, creating objects, network construction, etc.

The technote was designed to use a 1756-L85E processor, but an extra example has been added that uses a 1756-L75 v24 processor along with a 1756-EN2TR.

The example application provided was developed for a PanelView 800 2711R-T7T, but the terminal type can be changed inside Connected Components Workbench to match the hardware available as long as it is compatible.

The network used for this example is as follows:



ControlLogix processors do not support natively Modbus protocol. However, through the use of sockets it is possible to create logic that allows the ControlLogix processor to communicate over Modbus TCP. The code used in this example comes from the file named "ModbusTCP_Slave_R102.ACD" in the Sample Code Library at:

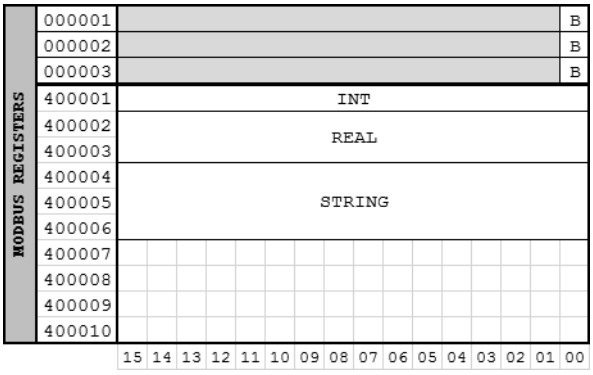
ControlLogix/CompactLogix Modbus TCP Program

The sample applications are provided and supported AS-IS. If additional Connections and or Transactions are required it is solely up to the programmer implementing the additional code to troubleshoot and support all of their modifications. **Rockwell Automation will not support the sample applications if any changes have been made to the code.**

The Sample Code Library link provides also a PDF document (ModbusTCP Slave Application R102.pdf) on how to configure the logic in Studio 5000, but for this example other than bringing the code as it was provided, the following tags were set:

- MBTU_Enable_MBTCP was set to 1 to enable the communications
- MBTU_EnetModulePort was set to sh1sh2 as the processor sits on slot 2 of the chassis

As a good practice, for ease of visualization, it is recommended to create a map of the tags that are going to be used in the project. For this example, the mapping will be as follows:



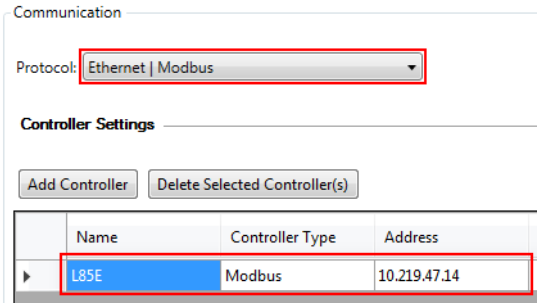
In total, 6 words of the 4xxxx registers will be mapped along with 3 bits of the 0xxxx registers, with the following breakdown:

- 1 x Boolean tag (1 bit) at register 000001.
- 1 x Boolean tag (1 bit) at register 000002.
- 1 x Boolean tag (1 bit) at register 000003.
- 1 x Unsigned Integer tag (16 bits) at register 400001.
- 1 x Real tag (32 bits) from register 400002 to 400003.

Top

- 1 x 6 characters string tag (8 bits per character = 48 bits) from register 400004 to 400006.

The following shows the necessary settings for the PanelView 800 Modbus TCP configuration:



(The name and IP can be changed to match your application.)

And this shows the configuration of the tags as per the previous mapping:

Tag Name	Data Type	Address	Controller
BOOL_01	Boolean	000001	L85E
BOOL_02	Boolean	000002	L85E
BOOL_03	Boolean	000003	L85E
FLOAT_01	Real	400003	L85E
STRING_01	String	400005.6H	L85E
WORD_01	Unsigned 16 bit integer	400002	L85E

Note that the Real and String tags point to the first register from the multiple that they use. Also, for the String tag the length and byte order needs to be specified. In this case, 6 characters are being used with a HiLo byte order and hence the .6H at the end. If LoHi byte order is needed it would be an L at the end instead. Keep in mind that since Real and String tags occupy several words, extra logic on the controller side is required to reconstruct the information from those tags.

Attachments

File

ModbusTCP Slave Application R102.pdf

File

PV800_L85E_Modbus.ccwarc

File

PV800_L85E_Modbus.cha

File

PV800_L85E_Modbus.ACD

Was this answer helpful?

○ **Yes** ○ **No**

RATE CONTENT

DISCLAIMER

Top

www.rockwellautomation.com

Top