TCPOPEN

Modicon X80 PMEUCM0302 Application

PMEUCM0302

Overview

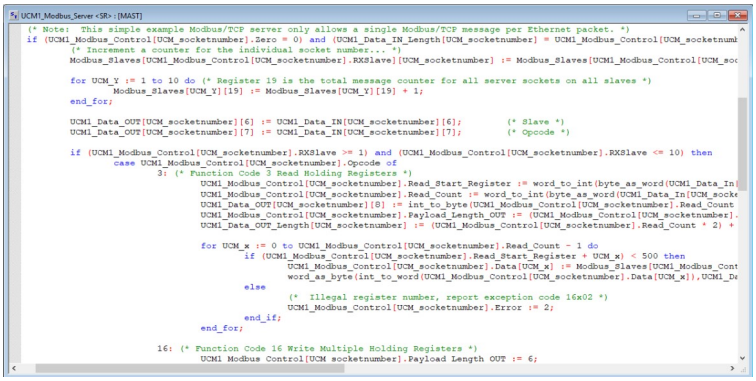
Niobrara's PMEUCM is a programmable communications module for Schneider Electric's Modicon M580 PAC. The TCPOPEN application for the PMEUCM allows the M580 CPU to control up to 16 Ethernet sockets as either Client or Server Available with two RS-232 serial ports and two Ethernet ports, the PMEUCM may allow the M580 PLC to communicate with a variety of external devices. The PMEUCM uses the Ethernet backplane to communicate with the M580 CPU or eNOC DIO master. Configuration is done through a DTM in Unity Pro. A LCD display and joystick make front panel troubleshooting quick and easy.

TCP and UDP Sockets

The TCPOPEN application allows PLC code to control each socket which may be configured to be a server (listen) or a client (connect). Settings such as local and remote TCP (or UDP) port numbers, IP Addresses, connection status, are all controlled through the IODDT structure provided by the DTM. DFBs are provided to automatically control moving the socket data to/from arrays of bytes for each socket.

Sample M580 ST Code

Sample Unity Pro code is provided to demonstrate a Modbus/TCP Server, Modbus/TCP Client, and Telnet Server that is written in structured text. The M580 Modbus/TCP server provides Holding Register data for 10 virtual slaves with FC3 reads and FC16 writes.

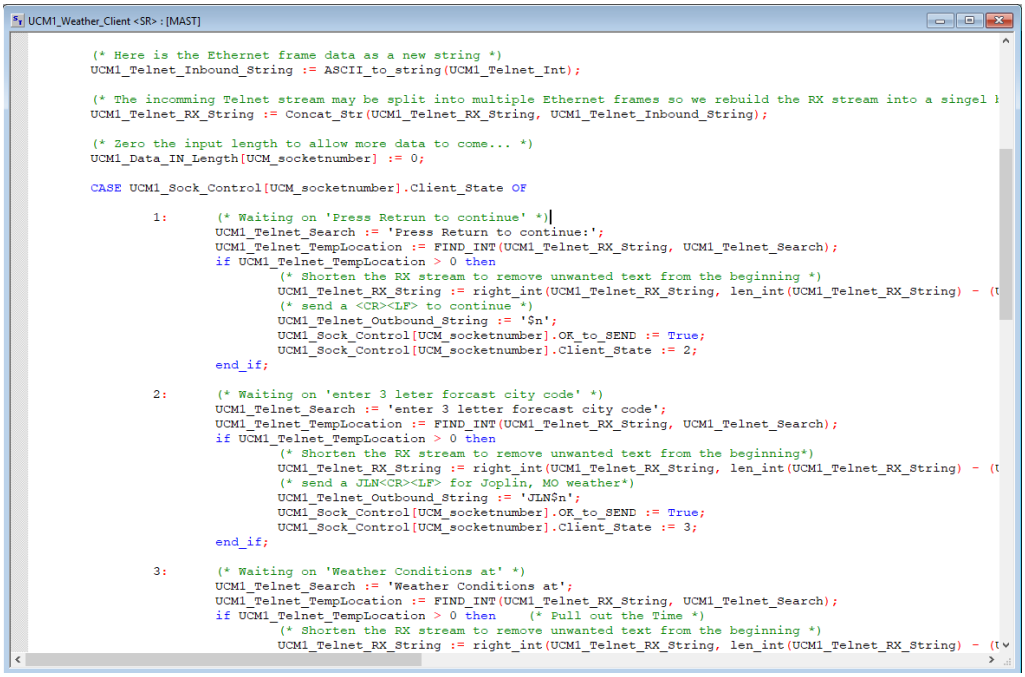


```
(* Note:  This simple example Modbus/TCP server only allows a single Modbus/TCP message per Ethernet packet. *)
if (UCM1_Modbus_Control[UCM_socketnumber].Zero = 0) and (UCM1_Data_IN_Length[UCM_socketnumber] = UCM1_Modbus_Control[UCM_socketnumb
        (* Increment a counter for the individual socket number... *)
        Modbus_Slaves[UCM1_Modbus_Control[UCM_socketnumber].RXSlave][UCM_socketnumber] := Modbus_Slaves[UCM1_Modbus_Control[UCM_soc

        for UCM_Y := 1 to 10 do (* Register 19 is the total message counter for all server sockets on all slaves *)
            Modbus_Slaves[UCM_Y][19] := Modbus_Slaves[UCM_Y][19] + 1;
        end_for;

        UCM1_Data_OUT[UCM_socketnumber][6] := UCM1_Data_IN[UCM_socketnumber][6];        (* Slave *)
        UCM1_Data_OUT[UCM_socketnumber][7] := UCM1_Data_IN[UCM_socketnumber][7];        (* Opcode *)

        if (UCM1_Modbus_Control[UCM_socketnumber].RXSlave >= 1) and (UCM1_Modbus_Control[UCM_socketnumber].RXSlave <= 10) then
            case UCM1_Modbus_Control[UCM_socketnumber].Opcode of
                3: (* Function Code 3 Read Holding Registers *)
                    UCM1_Modbus_Control[UCM_socketnumber].Read_Start_Register := word_to_int(byte_as_word(UCM1_Data_In[
                    UCM1_Modbus_Control[UCM_socketnumber].Read_Count := word_to_int(byte_as_word(UCM1_Data_In[UCM_socke
                    UCM1_Data_OUT[UCM_socketnumber][8] := int_to_byte(UCM1_Modbus_Control[UCM_socketnumber].Read_Count
                    UCM1_Modbus_Control[UCM_socketnumber].Payload_Length_OUT := (UCM1_Modbus_Control[UCM_socketnumber].
                    UCM1_Data_OUT_Length[UCM_socketnumber] := (UCM1_Modbus_Control[UCM_socketnumber].Read_Count * 2) +

                    for UCM_x := 0 to UCM1_Modbus_Control[UCM_socketnumber].Read_Count - 1 do
                        if (UCM1_Modbus_Control[UCM_socketnumber].Read_Start_Register + UCM_x) < 500 then
                            UCM1_Modbus_Control[UCM_socketnumber].Data[UCM_x] := Modbus_Slaves[UCM1_Modbus_Cont
                            word_as_byte(int_to_word(UCM1_Modbus_Control[UCM_socketnumber].Data[UCM_x]),UCM1_Da
                        else
                            (* Illegal register number, report exception code 16#02 *)
                            UCM1_Modbus_Control[UCM_socketnumber].Error := 2;
                        end_if;
                    end_for;

                16: (* Function Code 16 Write Multiple Holding Registers *)
                    UCM1_Modbus_Control[UCM_socketnumber].Payload_Length_OUT := 6;
```


Niobrara Research & Development Corporation
PO Box 3418
Joplin, MO 64803
800-235-6723
Tel:+1 417-624-8918
Fax:+1 417-624-8920
www.niobrara.com

The Modbus/TCP client code shows how to connect to remote servers and generate messages to do Modbus/TCP queries. The telnet server shows a simple server interface that can be accessed from any telnet client.

A second example is more complicated and involves connecting through the Internet to the Weather Underground text server. The client connects to the server, parses the prompt messages and sends replies to gather the local weather conditions based on a USA airport code.



The weather example deals with stream data from the remote server spanning multiple Ethernet frames and gives the user some idea of the power available to the M580 source code.

Other Use Examples

Some ideas implemented by customers using TCPOPEN in the PMEUCM:

- Connecting to a Printer Server using TCP sockets to send barcode print strings
- Reading data from a Positing System Radar using UDP sockets
- Sending HTTP Post messages to a web server in an IP Speaker to play mp3 files
- Communicating with a remote serial device connected to a terminal server

Niobrara Research & Development Corporation
PO Box 3418
Joplin, MO 64803
800-235-6723
Tel:+1 417-624-8918
Fax:+1 417-624-8920
www.niobrara.com