Motion Control Products

Application note
Host Comms Protocol to Modbus TCP Gateway

AN00228

Rev B (EN)

Utilise the flexibility of the AC500 PLC range to provide support for legacy Baldor serial communication protocols when using motion products without serial connectivity

Introduction

Since the introduction of the very first Mint based motion products, over 25 years ago, these products have supported a very simple ASCII based protocol (called Host Comms Protocol, or HCP for short) as a means of exchanging numerical data between two serial based systems. This protocol was enhanced in the past to support additional controllers and data locations and released as an additional protocol called Host Comms Protocol 2.
Details on these protocols can be found in application notes AN00110 and AN00129 available for download from the support area of new.abb.com/motion.

The MotiFlex e180 and MicroFlex e190 AC servo drives have no serial port, but it may be necessary to use this product as part of an update to an existing Flex+Drive^{II}, MintDrive^{II} or even MotiFlex e100 based solution where the serial port is being used to allow the drive to communicate with another system component via HCP.

This can be achieved by using the AC500 PLC to act as a gateway between the serial protocol and Modbus TCP which would be connected to the MotiFlex e180 or MicroFlex e190 drive's generic Ethernet port. The diagram below illustrates this principle (in this example we have assumed a MotiFlex e180 is shown but this could be a MicroFlex e190 also):

HCP client (e.g. PC, HMI, 3rd party PLC etc..)

Serial connection (RS232 or RS485)

AC500 PLC
PM554-T-ETH for RS485
PM573-ETH for RS232

Ethernet connection

MotiFlex e180
Modbus TCP server @
192.168.0.x

The PLC acts as a server to the incoming HCP message frames, decodes these, reads/writes the corresponding Netdata locations in the appropriate MotiFlex e180 or MicroFlex e190 drive and then sends a serial response back to the HCP client. So for example, if the HCP client (e.g. a third party PC or microcontroller) sends a request to write the value 6.2 to Comms location 9 on serial node address 5 the PLC will decode this, write 6.2 to Netdata location 9 on a MotiFlex e180 or MicroFlex e190 connected via Ethernet at IP address 192.168.0.5 and then (if the Modbus TCP write is successful) the PLC sends an "ACK" response to the HCP client.

Sample code for both the PM554 (for RS485 connections) and PM573 (for RS232 connections) PLCs is included with this application note.

PLC program features

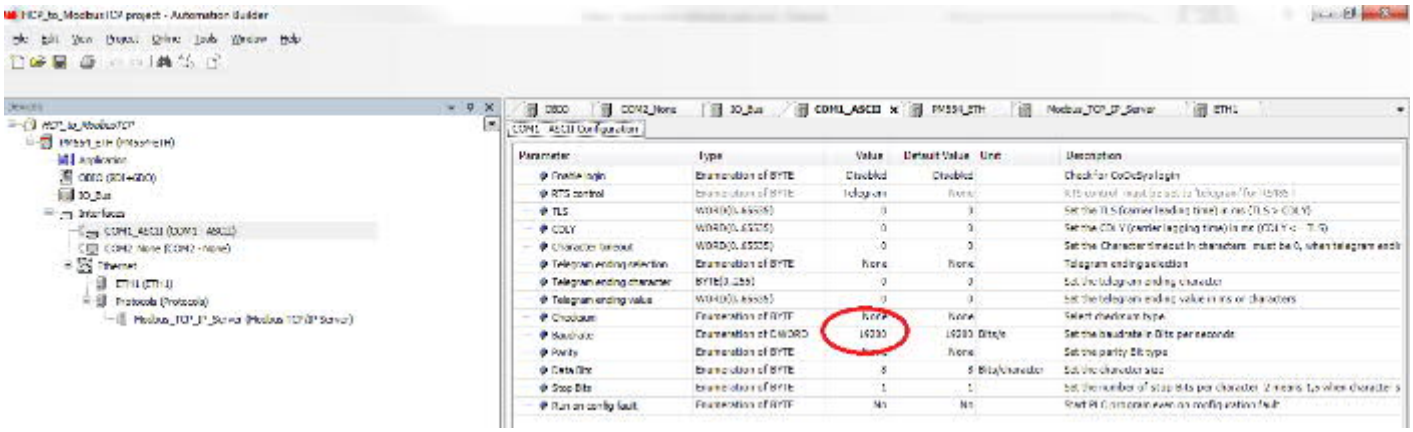
The PLC program samples provided should work in the majority of applications without the need for any modifications to the coded logic (see following section for details of some configuration options). Be sure to set the standard Ethernet address of any connected drives to match the required HCP serial node address (e.g. if the HCP client expects to communicate with serial node 2, set the drive's IP address to 192.168.0.2).
If using a RS485 (2 wire) serial connection please select the PM554-T-ETH PLC (if using a 4 wire RS422 serial port on the HCP client device check with the supplier whether this can work in two wire mode by linking TX+ with RX+ and linking TX- with RX-).
If using a RS232 serial connection please select the PM573-ETH PLC (the sample project expects COM1 to be used for this, but details on how to modify the PLC code to suit COM2 are included later in this document – be sure to wire RTS on the PLC to CTS on the HCP client and vice versa).

A single HCP transaction takes approximately 50-60ms to complete. The PLC is capable of supporting up to 10 Modbus TCP devices simultaneously (for more than 10 nodes multiple PLCs should be utilised). The addresses do not have to be sequential, so for example, it is possible to have drives connected to simulate serial nodes 2, 6 and 13. The gateway is also able to process HCP writes that include multiple data elements.

The PM554 solution makes use of the onboard digital outputs to provide some simple diagnostic information. Digital output 4 turns ON if a Modbus TCP write (and hence HCP write) has been performed successfully. Digital output 5 turns ON if a ModbusTCP read (and hence HCP read) has been performed successfully. These outputs will turn OFF if the relevant Modbus TCP transaction fails for some reason (e.g. Ethernet device disconnected). These diagnostic outputs are not included for the PM573-ETH solution as it has no onboard I/O.

PLC program configuration

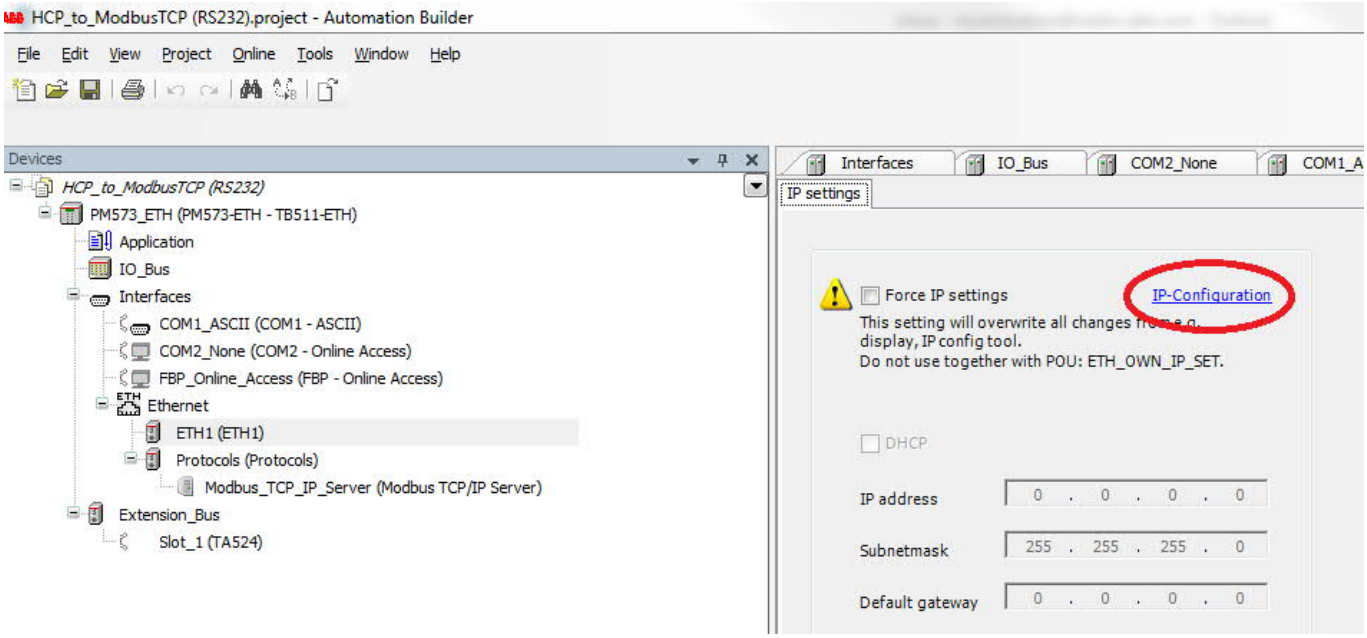
The sample application for RS485 (PM554 processor) is coded for serial port operation at 19200 baud. If this needs to be changed use Automation Builder to modify the serial port settings. Open the PLC project and double-click the COM1_ASCII entry in the device tree. The configuration window in the right hand pane will show the current settings for the serial port as shown below…



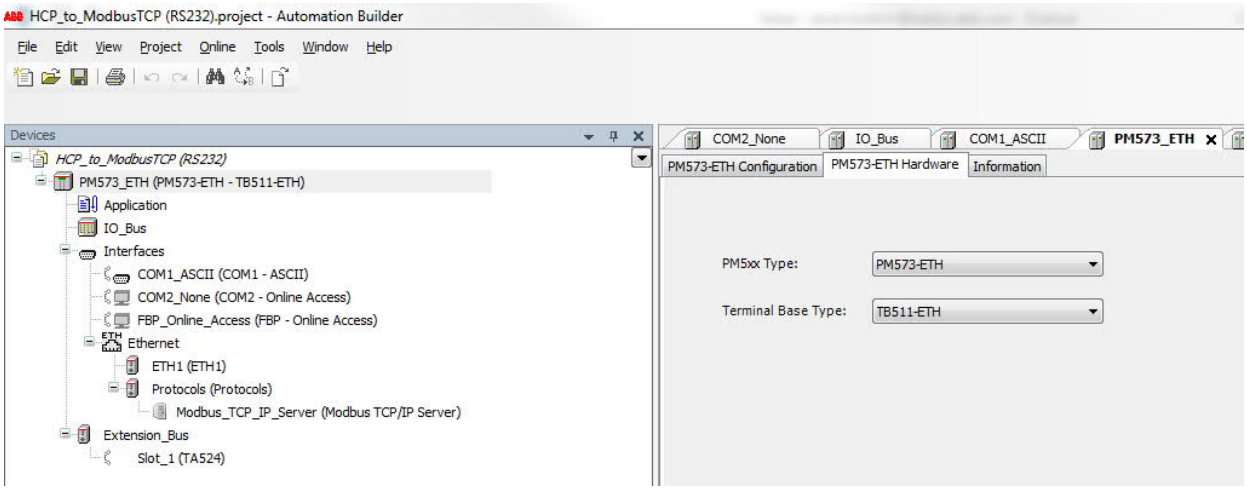
The screenshot has highlighted the baud rate setting which can be changed to suit the HCP client's settings as required (e.g. 57600 baud).

The sample application sets the OMB time for Modbus TCP (the time delay before an inactive Modbus TCP socket is closed) to 100ms (from the default value of 1000ms). This is to allow the PLC to handle 10 connected Ethernet nodes, with the default setting only 9 nodes are possible. If there are fewer than 10 nodes being used in the application and there is a requirement to speed the overall communication rate up very slightly then it is possible to increase this timeout to a value greater than the minimum time between transactions to a particular node (e.g. if the HCP client talks to node 4 every 500ms then setting the timeout to 1000ms would ensure the TCP socket is not closed and reopened for every transaction).

The PLC's default IP address is 192.168.0.10 so if the HCP client needs to address serial node 10 be sure to use Automation Builders IP configuration tool to change the PLC's IP address to a value not used by the HCP client (e.g. 192.168.0.200) because the drive connected on the Ethernet side of the PLC gateway will need to be assigned the address 192.168.0.10.

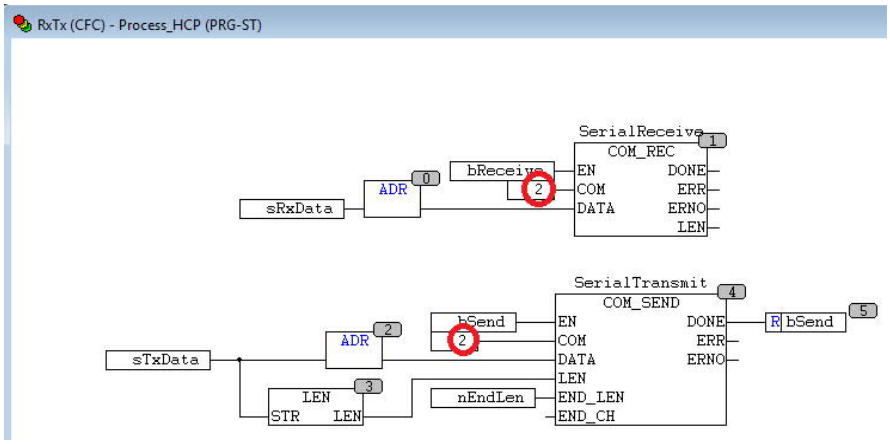


The sample project for the PM573-ETH (i.e. RS232) solution assumes a TB511-ETH base will be used in conjunction with the PM573 processor. If a different base is actually used, double-click the processor in the Automation Builder device tree, select the "PM573-ETH Hardware" tab from the right hand pane and then select a new 'Terminal Base Type' from the drop-down list…



The PM573-ETH project also assumes COM1 will be used (the serial port that uses the terminal strip connections), if the user prefers to use COM2 (the 9 way D-type connection) then use Automation Builder to change the operation mode for COM2 from Online Access to ASCII (and change COM1 back to Online Access). Be sure to set the setting for RTS control to 'RTS/CTS (DCE←DTE)'as it was for COM1 originally and set the Baud rate to suit the application requirements.

The application code for the RxtX action must also be edited via Codesys in this case to use COM2 as shown below…



As it is often easier to connect to terminal strips rather than D-type connectors it is anticipated that the default configuration for using COM1 will suit most applications.

Contact Us

For more information please contact your
local ABB representative or one of the following:

new.abb.com/motion
new.abb.com/drives
new.abb.com/drivespartners
new.abb.com/PLC