

# Modbus TCP Explicit Message Example: Read Register Request

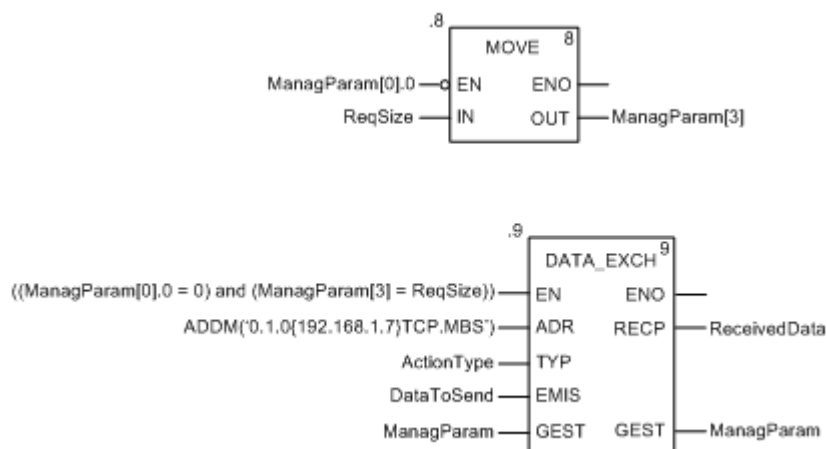

[About us](#)

## Overview

The following example shows you how to use the `DATA_EXCH` function block to send an explicit messaging **Modbus TCP** request to a remote device (at IP address 192.168.1.7) to read a single word located in the remote device at register 5391.

## Implementing the `DATA_EXCH` Function Block

To implement the `DATA_EXCH` function block, you need to create and assign variables for these blocks:



## Declaring Variables

In this example, the following variables were defined. You can, of course, use different variable names in your explicit messaging configurations.

Name	Type	Value	Comment	Used
ActionType	INT	16#01	Transmission followed by await reception	1
DataToSend	ARRAY[0..2] OF INT			2
DataToSend[0]	INT	16#1503	High byte of register address; Function Code	
DataToSend[1]	INT	16#000F	High byte of number of registers to read; Low byte of register address	
DataToSend[2]	INT	16#0001	Not used; Low byte of number of registers to read	
ManagParam	ARRAY[0..3] OF INT			6
ManagParam[0]	INT		High byte: Exchange #; Low byte: bit 1 = activity, bit 2 = cancel	
ManagParam[1]	INT		Operation Report; Communication Report	
ManagParam[2]	INT	2	Function block timeout = 2 (200 ms)	
ManagParam[3]	INT	6	Length of DataToSend parameter, in bytes	2
ReceivedData	ARRAY[0..1] OF INT			2
ReqSize	INT	5		3

## Configuring the Address Variable

The Address variable identifies the explicit message originating device (in this example, the communication module) and the target device. Note that the Address variable does not include the Xway address elements {Network.Station} because we are not bridging through another PLC station. Use the `ADDM` function to convert the following character string to an address:

`ADDM('0.1.0{192.168.1.7}TCP.MBS')`, where:

- rack = 0
- module (slot number) = 1
- channel = 0
- remote device IP address = 192.168.1.7
- message type = TCP
- protocol = Modbus

### Configuring the ActionType Variable

The ActionType variable identifies the function type for the DATA\_EXCH function block:

Variable	Description	Value (hex)
ActionType	Transmission followed by wait for response	16#01

### Configuring the DataToSend Variable

The DataToSend variable contains the target register address and the number of registers to read:

Variable	Description	Value (hex)
DataToSend[0]	<ul style="list-style-type: none"> <li>• High byte = Most significant byte (MSB) of register address 16#15 (21 decimal)</li> <li>• Low byte = function code: 16#03 (03 decimal)</li> </ul>	16#1503
DataToSend[1]	<ul style="list-style-type: none"> <li>• High byte = Most significant byte (MSB) of the number of registers to read: 16#00 (0 decimal)</li> <li>• Low byte = Least significant byte (LSB) of register address: 16#0F (15 decimal)</li> </ul>	16#000F
DataToSend[2]	CIP request instance information: <ul style="list-style-type: none"> <li>• High byte = not used: 16#00 (0 decimal)</li> <li>• Low byte = Least significant byte (LSB) of the number of registers to read: 16#01 (1 decimal)</li> </ul>	16#0001

**NOTE:** For detailed information about M580 network topologies, refer to the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures* and *Modicon M580 System Planning Guide for Complex Topologies*.

### Viewing the Response

Use a Unity Pro Animation table to display the ReceivedData variable array. Note that the ReceivedData variable array consists of the entire data buffer.

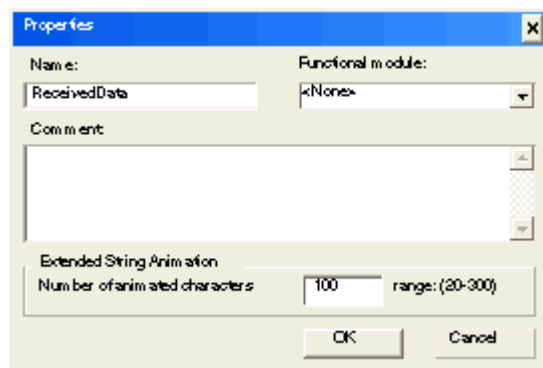
To display the Modbus TCP response, follow these steps:

Step	Action
1	In Unity Pro, select <b>Tools</b> → <b>Project Browser</b> to open the Project Browser.
2	In the Project Browser, select the <b>Animation Tables</b> folder, then click the right mouse button. A pop-up menu appears.
3	Select <b>New Animation Table</b> in the pop-up menu. A new animation table and its properties dialog both open.

4 In the Properties dialog, edit the following values:

<b>Name</b>	Type in a table name. For this example: <b>ReceivedData</b> .
<b>Functional module</b>	Accept the default <b>&lt;None&gt;</b> .
<b>Comment</b>	(Optional) Type your comment here.
<b>Number of animated characters</b>	Type in <b>100</b> , representing the size of the data buffer in words.

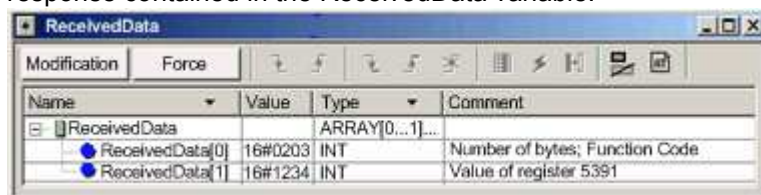
5 The completed **Properties** dialog looks like this:



Click **OK** to close the dialog.

6 In the animation table's **Name** column, type in the name of the variable assigned to the databuffer: **ReceivedData** and hit **Enter**. The animation table displays the ReceivedData variable.

7 Expand the ReceivedData variable to display its word array, where you can view the CIP response contained in the ReceivedData variable:



**Note:** Each array entry presents 2 bytes of data in little endian format. For example, '03' in word [0] is the low byte, and '02' is the high byte.