

Modbus[®] TCP for the Microchip TCP/IP Stack

*Author: Parthiv Pandya
Microchip Technology Inc.*

Modbus[®] is an industrial Transmission Control Protocol (TCP) used for remote monitoring and control applications. This application note describes a basic Modbus TCP server implementation and how it can be extended in a Modbus application.

The application software that accompanies this document was developed using Microchip's TCP/IP stack (v5.42) as a base framework. Although the software was developed using the PIC32MX360F512L microcontroller, it is easily ported to other PIC microcontrollers that support TCP/IP functionality.

OVERVIEW

This section provides an overview of the Modbus protocol. A detailed protocol specification is available from the Modbus Organization web site at: www.modbus.org.

Modbus was originally developed by Modicon Corporation (now Schneider Electric) as a means for communicating in a large industrial network over a twisted pair wire. The earlier versions of the protocol

were Modbus ASCII and Modbus RTU, which transmitted data over the RS-232 and RS-485 physical layers. Today, it is widely used over TCP/IP as Ethernet has become an industry standard for communication. One of the major advantages of using Ethernet is Internet connectivity.

The Modbus protocol can be defined as a client/server protocol where a client will request data, such as the temperature from a sensor or the status of a solenoid valve from a server, and the server will reply with this information. There can be one client with multiple servers or multiple clients with multiple servers within an implementation.

The information is saved in the server device within four tables. Two tables save information as single bit values known as Discrete Inputs and Coils (Discrete outputs). The other two tables save 16-bit values known as Holding registers and Input registers. Each table can contain up to 9999 members. Table 1 provides more information about these registers.

The Modbus protocol does not require that a manufacturer implement all of the primary tables.

Also, it is very common to find all of the I/O mapped only to the holding registers.

TABLE 1: MODBUS REGISTER TYPES

Primary Tables	Object Types	Access Mode	Description
Discrete Input	Single bit/Boolean values	Read-only	Used to represent sensor input.
Coil	Single bit/Boolean values	Read/Write	Used to represent output (i.e., value solenoid ON/OFF information).
Input Register	16-bit/Word	Read-only	Used to represent an analog input value or some other integer value.
Holding Register	16-bit/Word	Read/Write	Used to represent an analog output value or any requirement for holding data.

MODBUS[®] TCP FOR THE MICROCHIP TCP/IP STACK

MODBUS PROTOCOL

The protocol defines a simple application data unit (ADU), a Modbus packet format. The ADU consists of a Modbus application header (MBAP) and a protocol data unit (PDU).

TABLE 2: MBAP

Byte Number	Description
0-1	Transaction ID (TID) - Keeps track of Modbus packets.
2-3	Protocol ID (PID) - 0 for Modbus protocol.
4-5	Length of data to follow including the Unit Identification (UID), Function Code and the Data.
6	Unit Identification - Remote server ID.

TABLE 3: PDU

Byte Number	Description
7	Function Code (FCode) - Modbus command.
8-...	Data.

In the PDU, the number and type of data depends upon the Modbus command. The following example illustrates a Modbus transaction. Here, the client sends a request to read four holding register values starting from address 0x03, as shown in Table 4. The server replies with the four register values, as shown in Table 5.

Example 1: Read 4 holding registers with starting address 0x03 (0x03, 0x04, 0x05, and 0x06).

TABLE 4: REQUEST

0	1	2	3	4	5	6	7	8	9	10	11
00	01	00	00	00	06	01	03	00	03	00	04
TID		PID		Length		UID	FCode	Starting Address		Number of Registers	

TABLE 5: RESPONSE

0	1	2	3	4	5	6	7	8	9	10	11-18							
00	01	00	00	00	06	01	03	00	03	00	03	AB	00	00	05	45	AD	CA
TID		PID		Length		UID	FCode	Starting Address	Byte Count	Register Value								

MODBUS[®] TCP FOR THE MICROCHIP TCP/IP STACK

MODBUS TCP SERVER IMPLEMENTATION

To use the Modbus Server, add the `ModbusTCPServer.c`, `ModbusTCPServer.h` and replace the `MainDemo.c` files into your project.

To define a socket, add `#define TCP_PURPOSE_Modbus_TCP_SERVER N` (where N is the next number in the list) into the `#define TCP_SOCKET_TYPES` in the appropriate TCP/IP Configuration file. For example, this Modbus server uses the `XC32-EX16_ENC28` project from the Microchip TCP/IP stack as a base framework. The corresponding configuration file is `TCPIP_ENC28.h`.

To initialize the buffer for this socket, add `{TCP_PURPOSE_Modbus_TCP_SERVER, TCP_ETH_RAM, 200, 200}` in the `TCPsocketInitializer` structure.

MODBUS SERVER

Modbus uses port 502 for communication. This port number is defined using the `#define Modbus_PORT` macro in the `ModbusTCPServer.c` file.

The solution implements the following function codes:

- | | |
|---------------------------------|----|
| • ReadMultipleHoldingRegister | 3 |
| • ReadInputRegister | 4 |
| • WriteSingleCoil | 5 |
| • WriteMultipleHoldingRegisters | 16 |

The solution uses two buffers to receive and transmit Modbus commands and data. In addition to these buffers, the solution also uses two tables to save input registers and holding register values and a structure to save the Coil's address and its status (ON/OFF). These buffers and tables are described in the following two sections.

BUFFERS

`MODBUS_RX[MODBUS_RX_BUFFER_SIZE]`: This is a TCP buffer. It is used to receive TCP frames from the Modbus client. `MODBUS_RX_BUFFER_SIZE` is a macro that defines the buffer size. This value can be changed using `#define MODBUS_RX_BUFFER_SIZE` in the `ModbusTCPServe.h` files.

`MODBUS_TX[MODBUS_TX_BUFFER_SIZE]`: This buffer is used to transmit Modbus responses to the client. The buffer size can be changed using `#define MODBUS_TX_BUFFER_SIZE`.

TABLES

`HOLDING_REG[HOLDING_REG_SIZE]`: This table is used for the holding registers. The `HOLDING_REG_SIZE` macro can be used to change the number of registers used in the application.

`INPUT_REG[INPUT_REG_SIZE]`: This buffer is used for the input registers. The `INPUT_REG_SIZE` macro can be used to change the number of registers used in the application.

STATE MACHINE

The server employs a simple state machine that consists of two states:

- `SM_HOME`: Opens Modbus server socket
- `SM_RECEIVEDATA`: Receives Modbus requests from a client, formulates the response and replies with the requested data

In the `SM_HOME` state, the task attempts to open the Modbus server socket. This socket uses a `TCP_PURPOSE_MODBUS_TCP_SERVER` socket type.

Once the socket is successfully opened, the task function enters into the receiving mode where the server reads the received data from the Modbus buffer. The Modbus protocol message is decoded using the `MODBUS_COMMAND` structure inside the `ProcessReceivedMessage()` function. Depending on the function code, a simple switch statement formats the Modbus frame to respond to the client's request.

APPLICATION DEMONSTRATION

The application software demonstration implements 25 holding registers, 25 input registers, and 6 coils. It illustrates how to read and write constant values to holding registers. The software saves potentiometer and temperature sensor results into the first two locations of the input registers, and then sends them to a client upon request. The coils are used to control LEDs on a demonstration board.

MODBUS[®] TCP FOR THE MICROCHIP TCP/IP STACK

MODIFYING SOURCE CODE

1. In addition to the function codes used in the application, the `ModbusTCPServer.h` header file includes more function code definitions, such as:

- `ReadCoil` 1
- `ReadDiscreteInputs` 2
- `WriteSingleRegister` 6
- `WriteMultipleCoils` 15

After adding the definition, the user must add a switch statement in the `ModbusTCPServer.C` file. In the switch statement, add `ProcessReceivedmessage()` to decode the function code. Then, the user needs to develop a response function that will assemble the data packet, and send the response to the client.

2. `COIL` provides information about how to control an output pin. This can be used to control a solenoid or a switch. The `ProcessIO()` function provided in the `MainDemo.c` file contains the switch statement for the Coils' input output control. A user can modify this switch statement to add more coils.
3. Modbus uses big-endian representation for address and data information. However, some clients use little-endian representation. To support the latter, the solution uses flags, such as `ReadByteReverse`, `WriteByteReverse` and `InputByteReverse`, which decide whether the byte order change is required. According to the client's requirement, these flags can be set or reset.

CONCLUSION

This application note presents a very simple software solution for the Modbus TCP server.

REFERENCES

Microchip TCP/IP Stack

The Microchip TCP/IP stack can be downloaded from the Microchip web site at: www.microchip.com/tcpip. The stack Help file contains information on how to use Microchip's TCP/IP stack, the TCP/IP protocols, and the stack API.

Modbus Organization

The official Modbus protocol specifications are available from the Modbus Organization web site at: www.modbus.org

MODBUS[®] TCP FOR THE MICROCHIP TCP/IP STACK

APPENDIX A: SOURCE CODE

Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the "Company") is intended and supplied to you, the Company's customer, for use solely and exclusively with products manufactured by the Company.

The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

All of the software covered in this application note is available as a single WinZip archive file. This archive can be downloaded from the Microchip corporate Web site at:

www.microchip.com

MODBUS[®] TCP FOR THE MICROCHIP TCP/IP STACK

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC³² logo, rPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.


Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniclient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICTail, REAL ICE, rLAB, Select Mode, SQL, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 978-1-62077-503-5

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
= ISO/TS 16949 =

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://www.microchip.com/support>
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820