R-IN32M3, R-IN32M4 Series, RZ/T1 Group

TFTP Driver

R01AN3939EJ1000
Rev.1.00
July 11, 2017

Introduction

This manual provides middleware specifications and describes the outline and specifications of TFTP as well as hardware configuration, software configuration, and driver specifications to achieve functions.

Target Devices

- R-IN32M3-EC
- R-IN32M3-CL
- RIN32M4-CL2
- RZ/T1

Contents

1. Outline of TFTP

TFTP is a protocol to transfer files through the TCP/IP network like FTP.

The FTP establishes a connection between client and server through the TCP network to perform transmission processing. On the other hand, the TFTP performs processing without using the UDP connection. Though the TFTP does not use any connection, it uses acknowledgment to ensure communication reliability. This protocol is used to transfer small amount of data such as firmware update of router.

Differences from FTP

- FTP needs user name and password authentication , but TFTP does not need them.
- FTP runs on the TCP, but TFTP runs on the UDP. shows the positioning of TFTP.

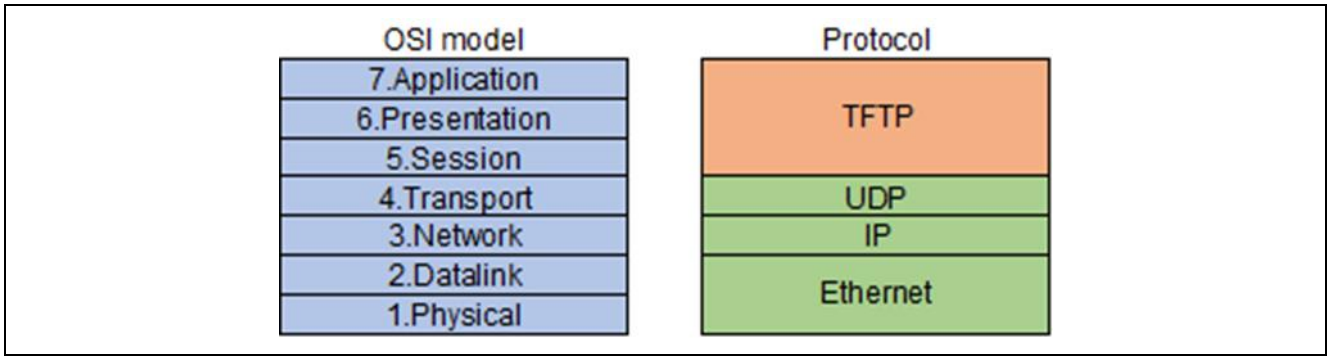


Figure 1-1 OSI Model and TFTP Protocol

1.1 Packet Format

Table 1-1 shows packets handled in the TFTP.

This middleware supports packets other than read request (RRQ).

Table 1-1 Packets Handled in TFTP

Opcode	Operation	Support
1	Read request (RRQ: Read RQuest)	Unsupported
2	Write request (WRQ: Write RQuest)	Supported
3	Data (DATA)	Supported
4	Response (ACK: ACKnowledgment)	Supported
5	Error (ERROR)	Supported

As shown in Figure 1-2, the UDP payload composes a TFTP packet. The first 2 bytes are an operation code (opcode) that indicates the TFTP's packet type. The format of subsequent bytes varies from packet to packet.

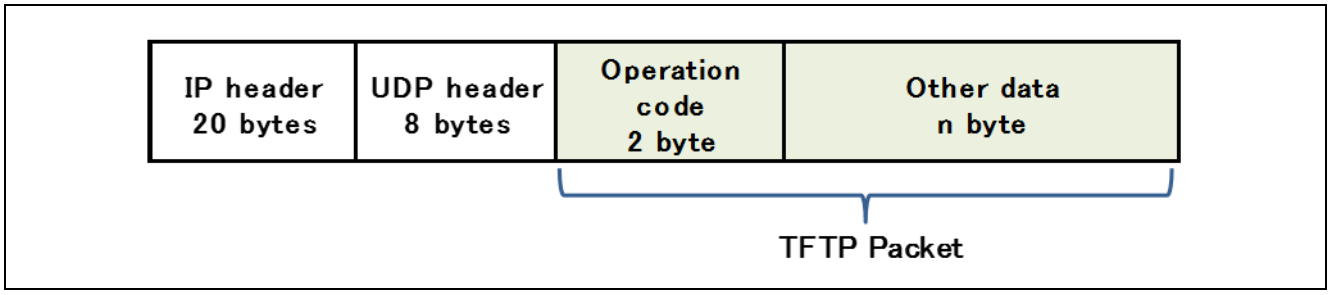


Figure 1-2 TFTP Packet Format

Formats of each TFTP packet are shown below.

■ WRQ packet format

This packet is used by clients to send files to the server.

Operation Code	File Name	Null	Mode	Null
2 bytes	n bytes (string)	1 byte	n bytes (string)	1 byte

· Operation Code: Opcode "2"

· File Name: File name to be updated

· Mode: Format of file to be transferred. One of the following mode names is specified using the ASCII code.

netascii … FTP's ASCII mode converted to the character code used at the reception side and stored

octet … FTP's binary mode sent from the transmission side without conversion

mail … Same as netascii mode. The name of mail receiving user is entered instead of file name.

■ DATA packet format

This packet is used to transfer files divided into blocks.

Operation Code	Block No	Data
2 bytes	2 bytes	n bytes

· Operation Code: Opcode "3"

· Block No: Block number

· Data: Consists of a 512-byte data size. The last block consisting of less than 512 bytes shows the end of the file.

■ ACK packet format

This packet is used to notify the other side that a DATA packet has been received.

This packet has a block number to report the received block number to the other side.

Operation Code	Block No
2 bytes	2 bytes

· Operation Code: Opcode "4"

· Block No: After a DATA packet is received, the received block number is returned.

After a WRQ packet is received, block number 0 is returned.

■ ERROR packet format

This packet is sent to the other side when an error occurs.

Operation Code	Error Code	Error Message	Null
2 bytes	2 bytes	n bytes	1 byte

· Operation Code: Opcode "5"

· Error Code: Error code

· Error Message: Error message

· Null: End of error message

* For error codes and messages, see Table 4-1 List of Macro.

1.2 Packet Sequence

1. The client sends WRQ to the server with a port number of 69. * Port 69 is used only when WRQ is received.

2. The server sends an ACK response with a block number of 0. * At this time, an ACK response is sent with a random port (not Port 69). The subsequent processing is performed with this port.

3. The client starts sending 512-byte DATA.

4. Each time the server receives DATA, it returns an ACK response with the same block number.

5. The client sends a DATA packet in the last block (0 to 511 bytes).

6. After the client receives the last ACK, the processing is completed.

7. If a packet is lost during transmission, retry processing is performed.

   If a DATA packet does not arrive within the specified time, the immediately previous ACK packet is resent as timeout processing. When no response is received after the set retry times, an error occurs.

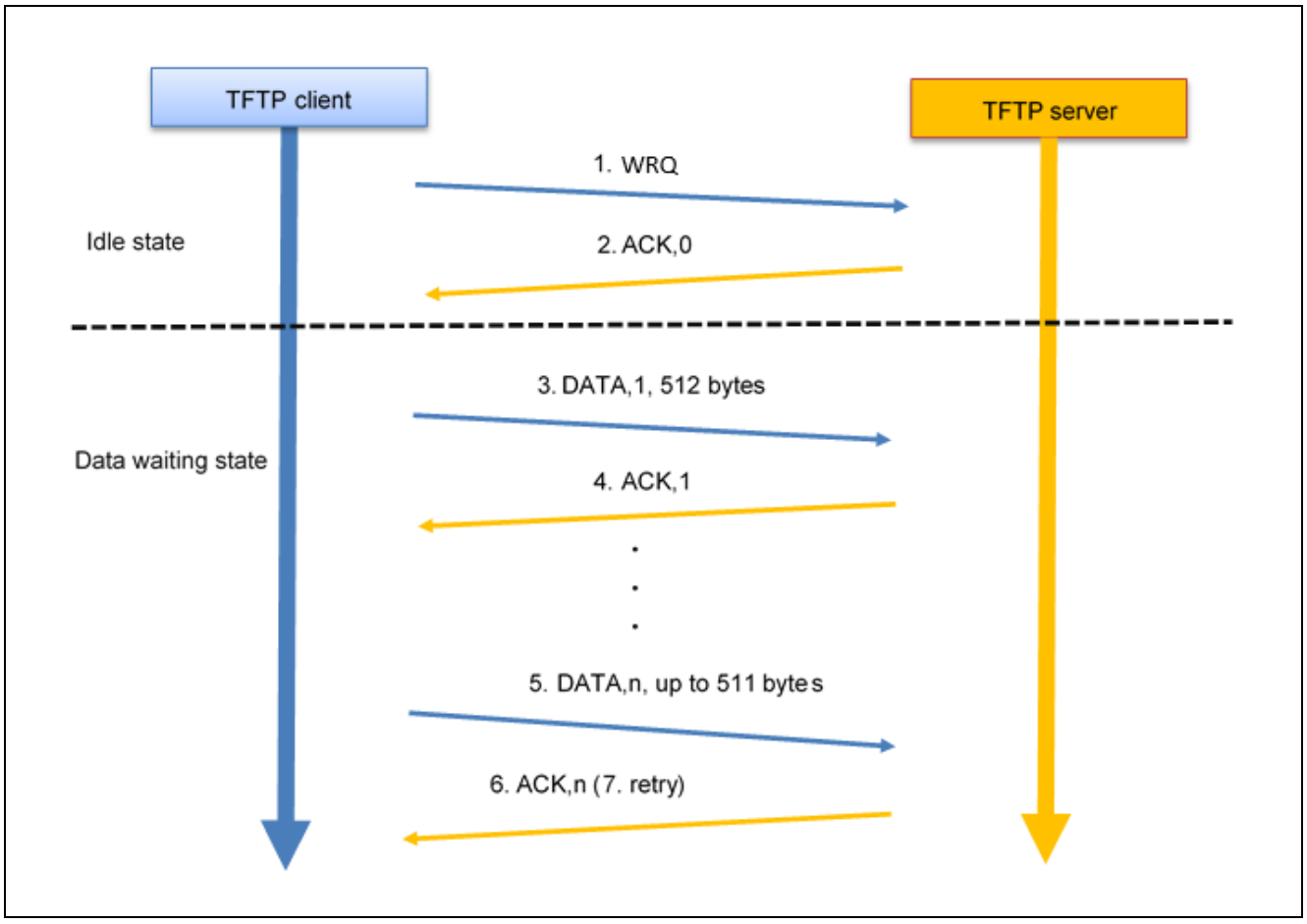


Figure 1-3 Packet Sequence

1.3 State Transitions

Table 1-2 shows state transitions of each event.

* DATA acceptance, time-out, and retry over in the idle state are not processed.

  If any unexpected event occurs, it is handled as an error.

Table 1-2 State Transitions

State / Event	Idle State	DATA Waiting State
WRQ acceptable	ACK response → DATA waiting state	ACK response → DATA waiting state
DATA acceptable	-	ACK response → DATA waiting state
DATA received	-	ACK response → Idle state
Time-out	-	ACK response → DATA waiting state
Retry over	-	Error → Idle state
Other than above	Error → Idle state	Error → Idle state

1.4 TFTP Server Flow

Figure 1-4 TFTP Server Flow, shows a simple flow of this middleware.

After initialization, this middleware decides the state, accepts DATA, and then accepts WRQ.

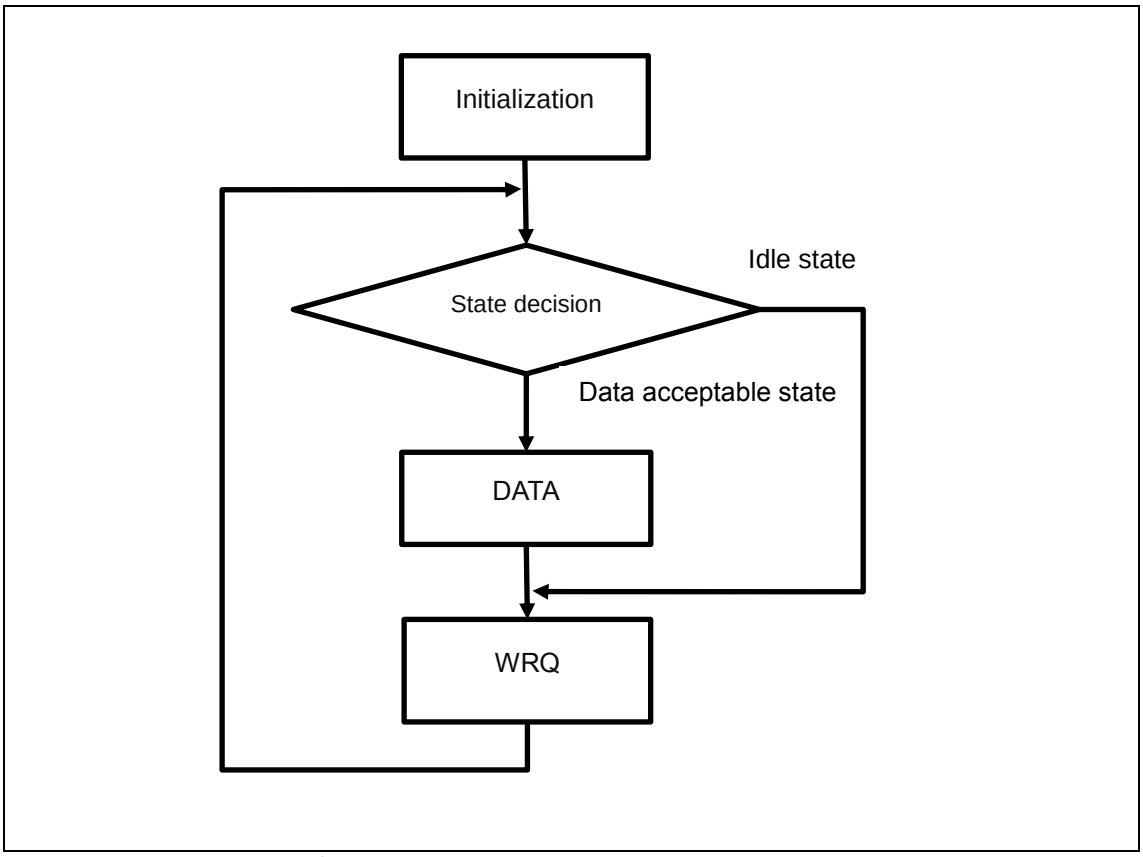
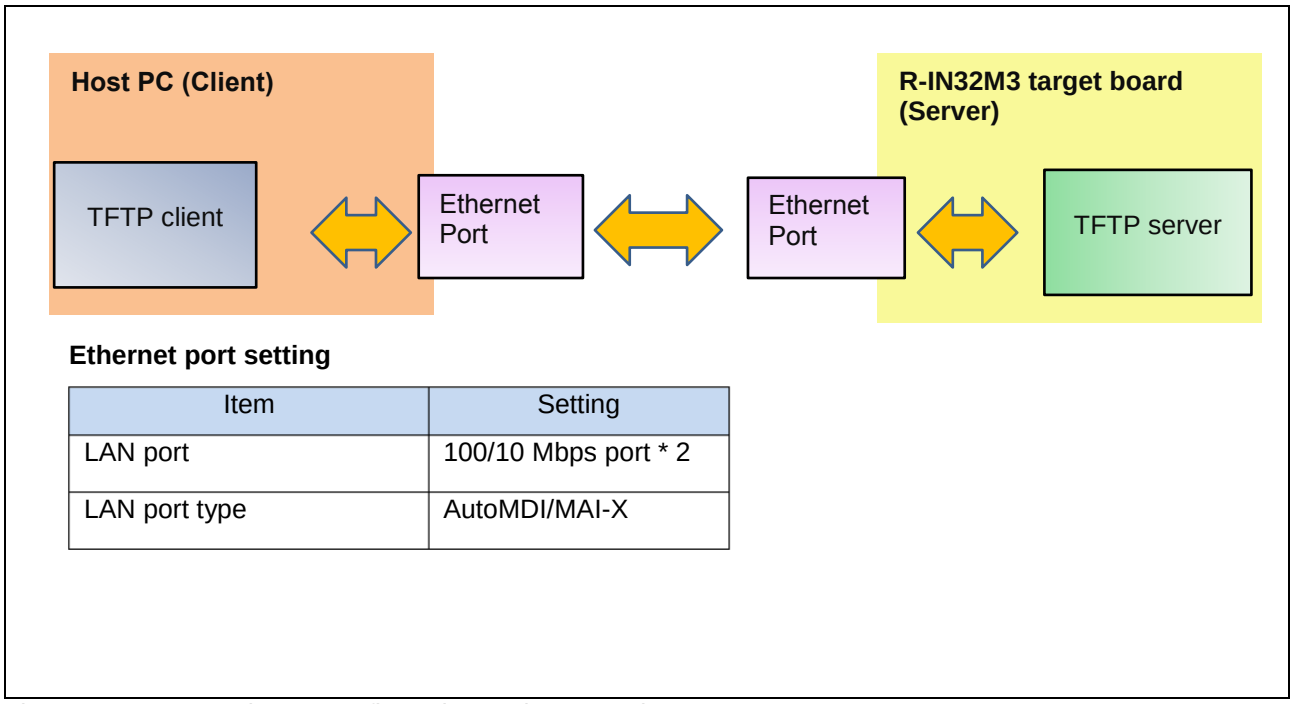


Figure 1-4 TFTP Server Flow

2. TFTP Server Middleware Configuration

2.1 Hardware Configuration

The client PC is connected to the R-IN32M3 target board through Ethernet communication.



Ethernet port setting

Item	Setting
LAN port	100/10 Mbps port * 2
LAN port type	AutoMDI/MAI-X

Figure 2-1 Hardware Configuration and Port Settings

2.2 Software Configuration

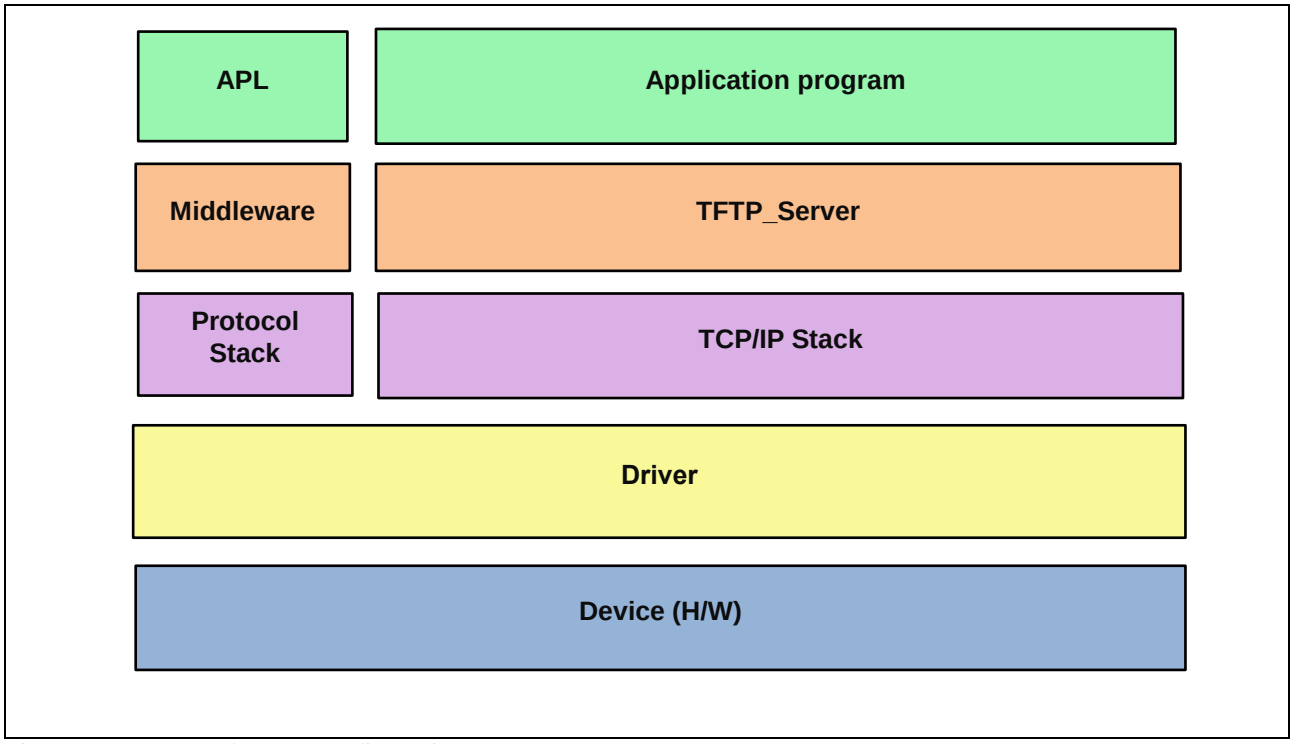


Figure 2-2 Software Configuration

3. Function Specifications

· Ack structure

```
typedef struct t_ack {
        uint16_t    op;             /*Operation code*/
        uint16_t    block;          /*Block number*/
}ack
```

· Err structure

```
typedef struct t_err {
        uint16_t    op;             /*Operation code*/
        uint16_t    errcd;          /*Error code*/
        char        msg[1500-4]     /*Message*/
}err
```

tftp_wrq_socket_init	WRQ socket initialization

[Format]

int tftp_wrq_socket_init(void);

[Parameter]

None

[Return value]

OK	Successful completion
API_ERR	API error

[Explanation]

This function initializes the socket for WRQ and the buffer that receives data from the client.

This function creates the socket of Port 69 that accepts connection from any address.

tftp_data_socket_init	DATA socket initialization

[Format]

int tftp_data_socket_init(void);

[Parameter]

None

[Return value]

OK	Successful completion
API_ERR	API error

[Explanation]

This function initializes the socket for DATA and the buffer that receives data from the client.

This function creates a socket for random port that accepts connection from any address.

tftp_cmd_wrq	WRQ reception processing

[Format]

int tftp_cmd_wrq(TFTP_PARAM_TABLE *param);

[Parameter]

TFTP_PARAM_TABLE	*param	Parameter table

[Return value]

OK	Successful completion
ERR	Error
API_ERR	API error
NO_DATA	No data received

[Explanation]

This function performs processing when the socket for WRQ receives data.

This function judges the received packet. If it is not a WRQ packet, an error packet is returned.

When it is a WRQ packet, this function returns an ACK response, acquires the file name and mode name from the packet, and then transfers them to the callback function.

When the processing has been successfully completed, this function creates a socket for DATA and enters the data acceptable state.

tftp_cmd_data	DATA reception processing

[Format]

int tftp_cmd_data(TFTP_PARAM_TABLE *param);

[Parameter]

TFTP_PARAM_TABLE	*param	Parameter table

[Return value]

OK	Successful completion
ERR	Error
API_ERR	API error

[Explanation]

This function performs processing when the socket for DATA receives data.

This function judges the received packet. If it is not a DATA packet, an error packet is returned. When it is a DATA packet, this function acquires the block number and data from the packet and then transfers them to the callback function.

If a DATA packet does not arrive within the specified time, the immediately previous ACK packet is resent as timeout processing.

This function judges the DATA packet size. When the DATA packet size is less than 512 bytes, data reception is completed and the callback function is called.

If WRQ is received during DATA reception, the processing transitions temporarily to WRQ judgment.

tftp_send_ack	ACK response processing

[Format]

int tftp_send_ack(int block);

[Parameter]

int	block	Block number

[Return value]

OK	Successful completion
API_ERR	API error

[Explanation]

This function is used to send an ACK response to the client. This function creates a packet from parameters and sends it to the client. Furthermore, this function decides state and switches sockets used for transmission.

tftp_send_err	ERR response processing

[Format]

int tftp_send_err(int errcd,const char *errmsg);

[Parameter]

int	errcd	Error code
const char	*errmsg	Error message pointer

[Return value]

OK	Successful completion
API_ERR	API error

[Explanation]

This function is used to return an error response to the client. This function creates a packet from parameters and sends it to the client. Furthermore, this function decides state and switches sockets used for transmission.

tftp_end	End processing

[Format]

void tftp_end(void);

[Parameter]

None

[Return value]

None

[Explanation]

This function is used when the middleware processing is exited with an error. This function closes the currently created socket.

4. Middleware Specifications

This middleware defines callback functions that operate according to requests from the client.

Therefore, this middleware is generally available for purposes other than firmware update.

* This middleware transfers predetermined parameters, performs arbitrary processing, and then receives a return value according to the specification.

· TFTP parameter structure

```
typedef struct tftp_param_table {
        Int (*wrq_cbk)(char *,char *)          WRQ reception callback function pointer
        Int (*data_cbk)(int ,uint8_t * ,int)   DATA reception callback function pointer
        void(*end_cbk)(void)                   Reception completion callback function pointer
}TFTP_PARAM_TABLE
```

[Explanation]

Functions to be used in the TFTP server are registered in this structure.

Table 4-1 List of Macros

Macro Name	Value	Message	Description
TFTP_OK	0		Successful completion
TFTP_ERR0	1	Not defined, see error message (if any).	0: Undefined See error messages.
TFTP_ERR1	2	File not found.	1: Was found.
TFTP_ERR2	3	Access violation.	2: Access violation
TFTP_ERR3	4	Disk full or allocation exceeded.	3: No writing area is left in the disk.
TFTP_ERR4	5	Illegal TFTP operation.	4: Illegal TFTP command
TFTP_ERR5	6	Unknown transfer ID.	5: Unknown transfer ID
TFTP_ERR6	7	File already exists.	6: A file already exists.
TFTP_ERR7	8	No such user.	7: The user is not defined.
TFTP_PORT_FIX_MODE	0	-	tftp fixed port mode switching 0 = disable, 1 = enable

wq_cbk WRQ callback function

[Format]

```
int wrq_cbk(char *filename ,char *mode);
```

[Parameter]

char	*filename	Pointer to WRQ packet parameter “filename”
char	*mode	Pointer to WRQ packet parameter “mode”

[Return value]

A return value is selected according to the list of macros.

[Explanation]

This function performs callback function processing when a WRQ packet is received.

This function performs arbitrary processing for parameters.

data_cbk	DATA callback function

[Format]

void data_cbk(int blockno ,uint8_t *buf,int size);

[Parameter]

int	blockno	DATA packet parameter “block”
uint8_t	*buf	Pointer to DATA packet parameter “data”
int	size	Valid size of “data”

[Return value]

None

[Explanation]

This function performs callback function processing when a DATA packet is received.

This function performs arbitrary processing for parameters.

end_cbk	END callback function

[Format]

void end_cbk(void);

[Parameter]

None

[Return value]

None

[Explanation]

This function performs callback function processing when reception of DATA is completed.

tftp_server	TFTP server

[Format]

int tftp_server(TFTP_PARAM_TABLE *param);

[Parameter]

TFTP_PARAM_TABLE	*param	Callback function structure

[Return value]

None

[Explanation]

This function initializes the TFTP server and accepts requests from the TFTP client.

This function decides the idle state and the data acceptable state and switches acceptance processes.

Example of implementation

```
int wrq_cbk(char *filename , char *mode){
        return 0;
}

int data_cbk(int blockno , uint8_t *buf,int size){
        return 0;
}

int end_cbk(void){
        return 0;
}

int main()
{
        TFTP_PARAM_TABLE param;
        int result;

        pram.wrq_cbk = wrq_cbk;
        param.data_cbk = data_cbk;
        param.end_cbk = end_cbk;

        re = tftp_server(&param);
        if(re == OK){
                /*Successful completion*/
        }

}
```

4.1 Setting TFTP Fixed Port Mode

The fixed port mode specifying macro “TFTP_PORT_FIX_MODE” can switch port settings when a response is returned to the client. Random port setting is made by default.

Table 4-2 TFTP Fixed Port Mode

Macro Name	Definition File	Value	Description
TFTP_PORT_FIX_MODE	tftp_server.h	0	Random port
		1	Port 69 fixed mode

Supplementary Note: When using Windows standard TFTP client, use the fixed port mode.

5. Sample Software Tutorial

This package is sample software for the R-IN32M3.

The TFTP middleware can be used in common with products of the R-IN32M3 and R-IN32M4 series and the RZ/T1 group.

This section describes the procedures for running the sample software and confirming operation by using the Windows standard TFTP client.

Outline of operation:

When the sample software is run, file transfer is accepted as a TFTP server.

The desired processing can proceed at any of the following times.

- When a WRQ packet is received
- When a DATA packet is received
- When reception of a DATA packet is completed

5.1 Preparation

5.1.1 Integrating the Files

This package consists of files which are not included in the R-IN32M3 Series TCP/IP Stack.

The TFTP middleware and sample software become available by integrating this package with the R-IN32M3 Series TCP/IP Stack.

Obtain the R-IN32M3 Series TCP/IP Stack from the sample code on the R-IN32M3 product page and write the files of this package to it.

Figure 5-1 shows the hierarchy of folders and files after writing of the files and folders of this package.

The folders and files of this package to be newly added to the R-IN32M3 Series TCP/IP Stack are shown in red text.

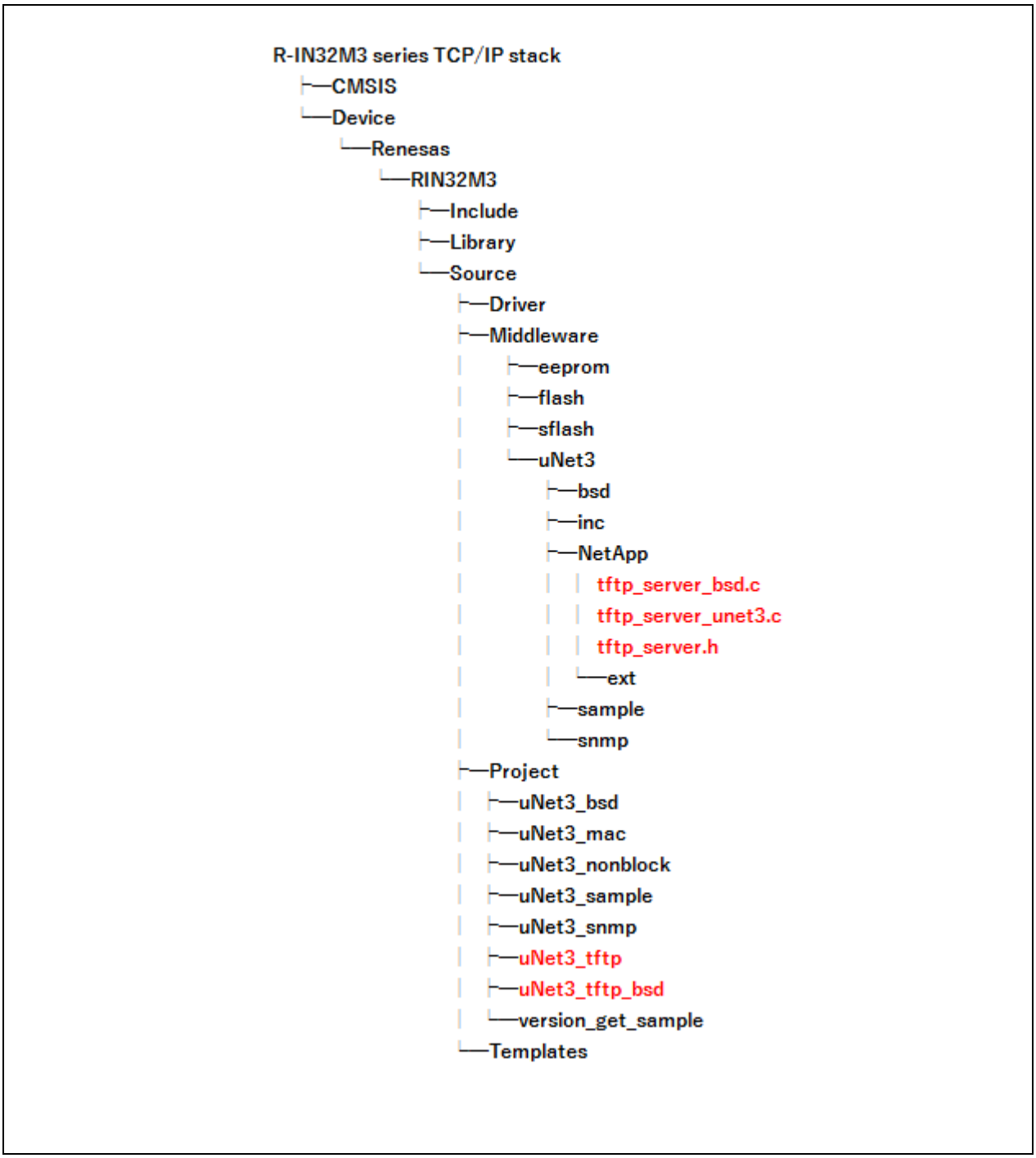


Figure 5-1 Configuration of the TFTP Folders

5.1.2 Connecting the Board to the Host PC

Board to be used: TS-R-IN32M3-EC R-IN32M3-EC Evaluation Board

Connect an Ethernet cable to LAN port 1 by following the procedure described in section

, Hardware Configuration.

5.1.3 IP Address Settings

The settings for the IP address of the R-IN32M3 board are as follows.

Subnet mask: 255.255.255.0

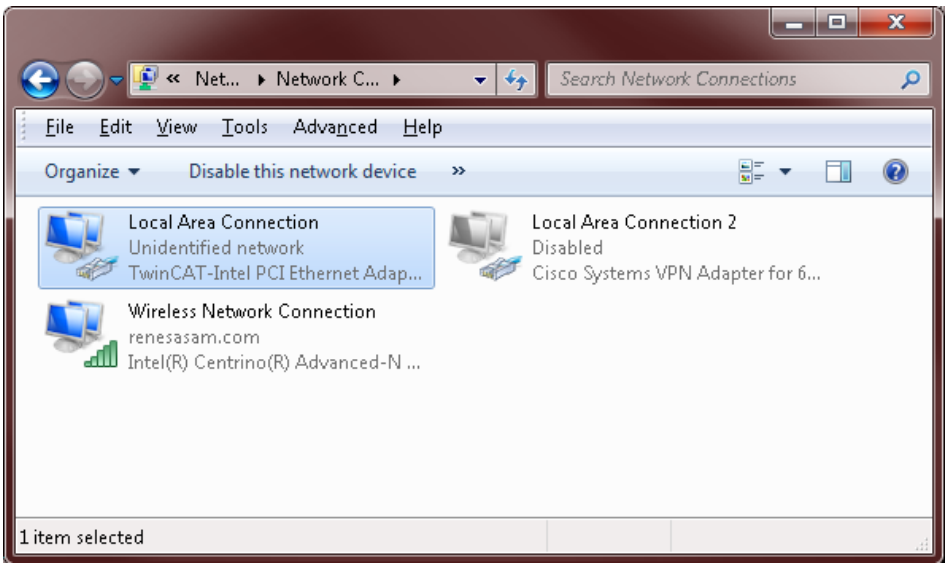
IP address: 192.168.1.100

Set the IP address of the host PC in the same section of the network as that of the R-IN32M3 board.

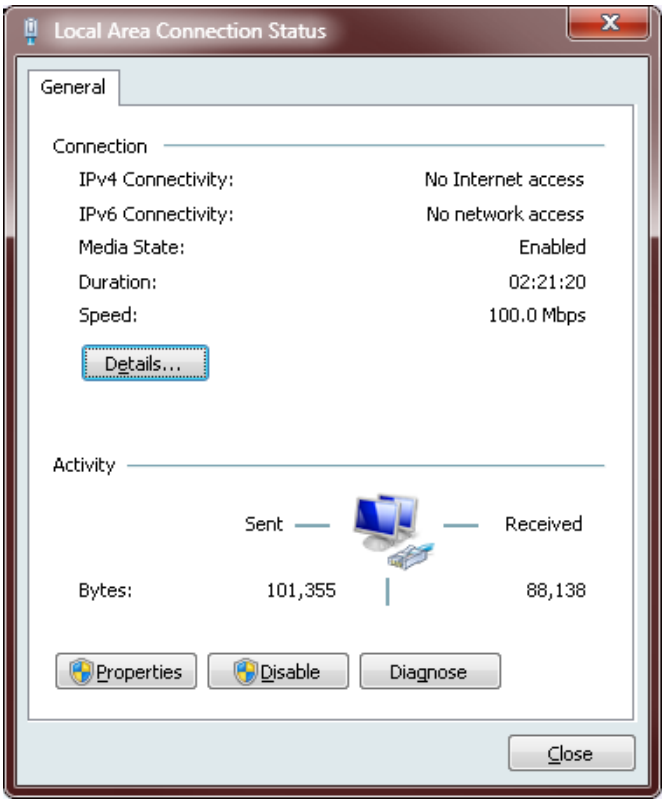
The following is the procedure for setting the IP address.

- Open [Network Connections].

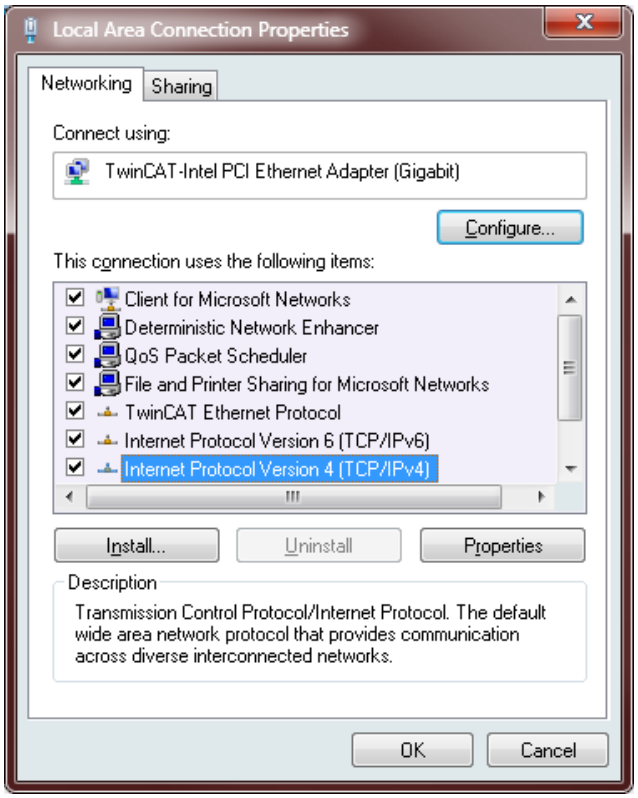
  In Windows 7, go to Control panel -> Network and Sharing Center -> Change adapter settings.



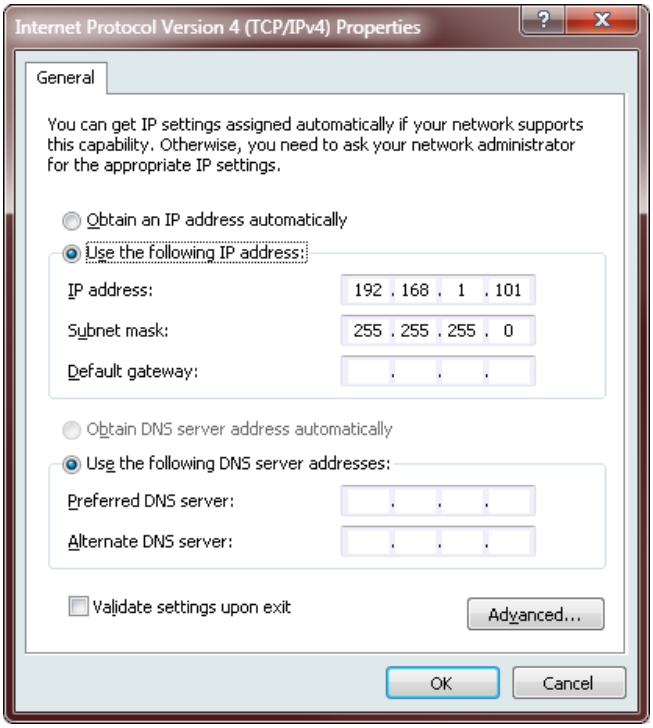
- Double-click on [Local Area Connection] and press the [Properties] button in the [Local Area Connection Status] dialog box.



• Select [TCP/IPv4] and press the [Properties] button in the [Local Area Connection Properties] dialog box.



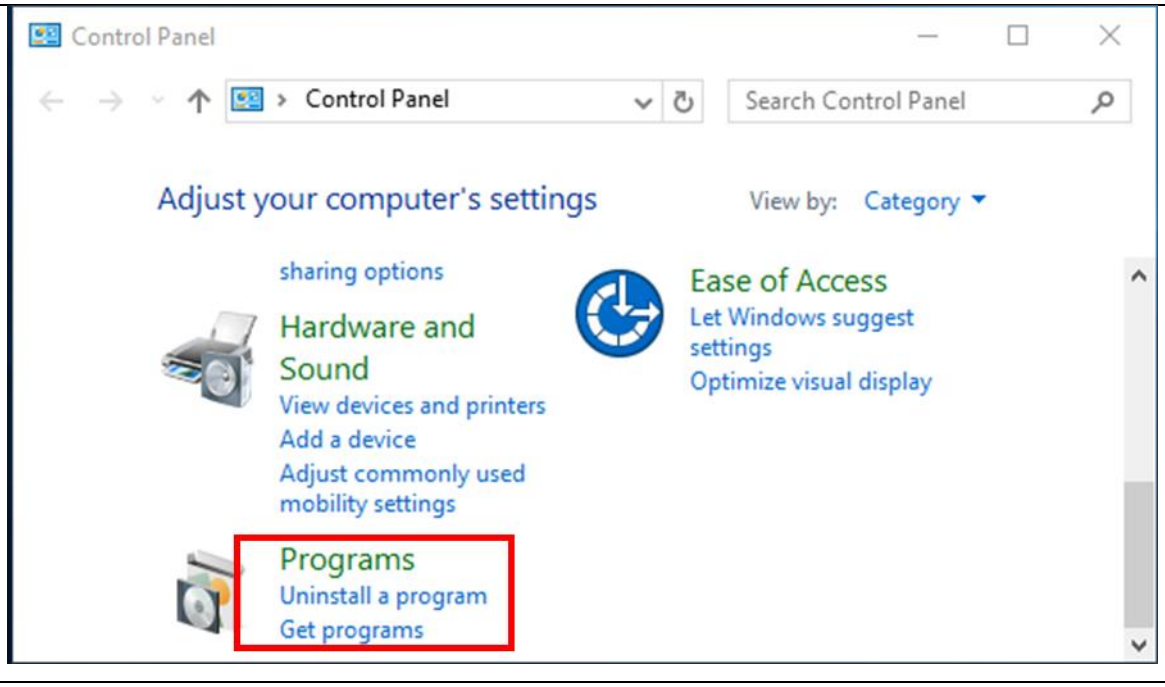
• Set [IP address] to “192.168.1.101” and [Subnet mask] to “255.255.255.0”.



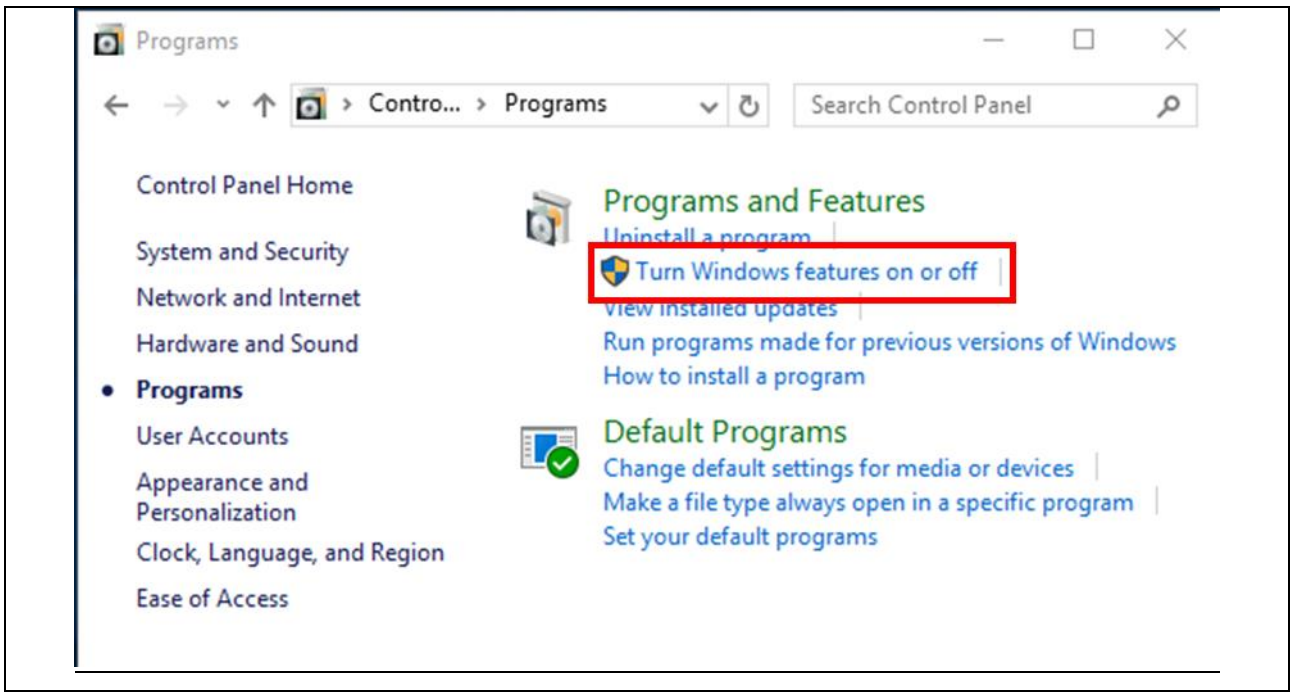
5.1.4 Enabling the Standard Windows TFTP Client

Depending on the terminal, the standard Windows TFTP client may be disabled by default. Therefore, the following describes how to make the setting to enable it.

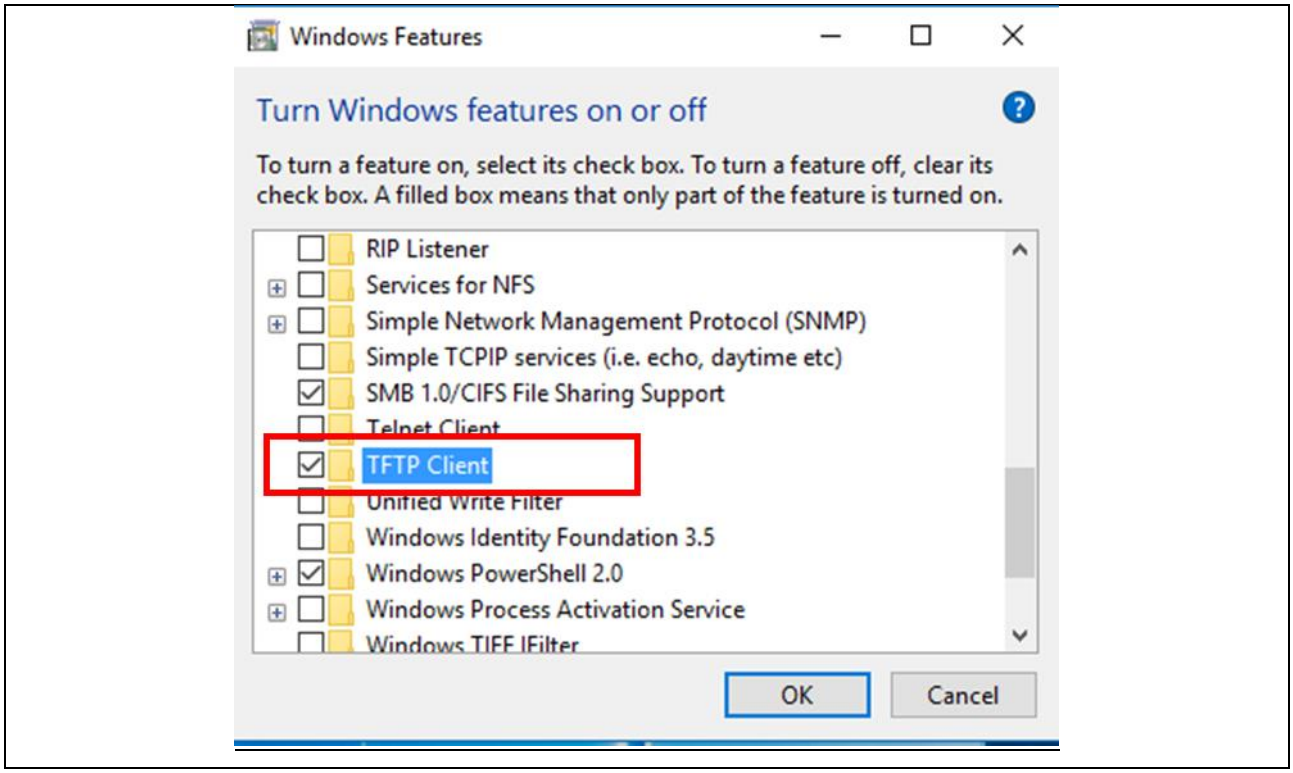
• Open [Control Panel] and select [Programs].



• In [Programs and Features], select [Turn Windows Features On or Off].



• Check the [TFTP client] box from among the displayed items.
After that, click on [OK] to enable the client.



5.1.5 Running the Sample Software

Set the value of the TFTP_PORT_FIX_MODE macro defined in \Device\Renesas\RIN32M3\Source\Middleware\uNet3\NetApptftp_server.h to “1” (fixed port mode).

Run the sample software from either of the following workbench files.

Workbench files:

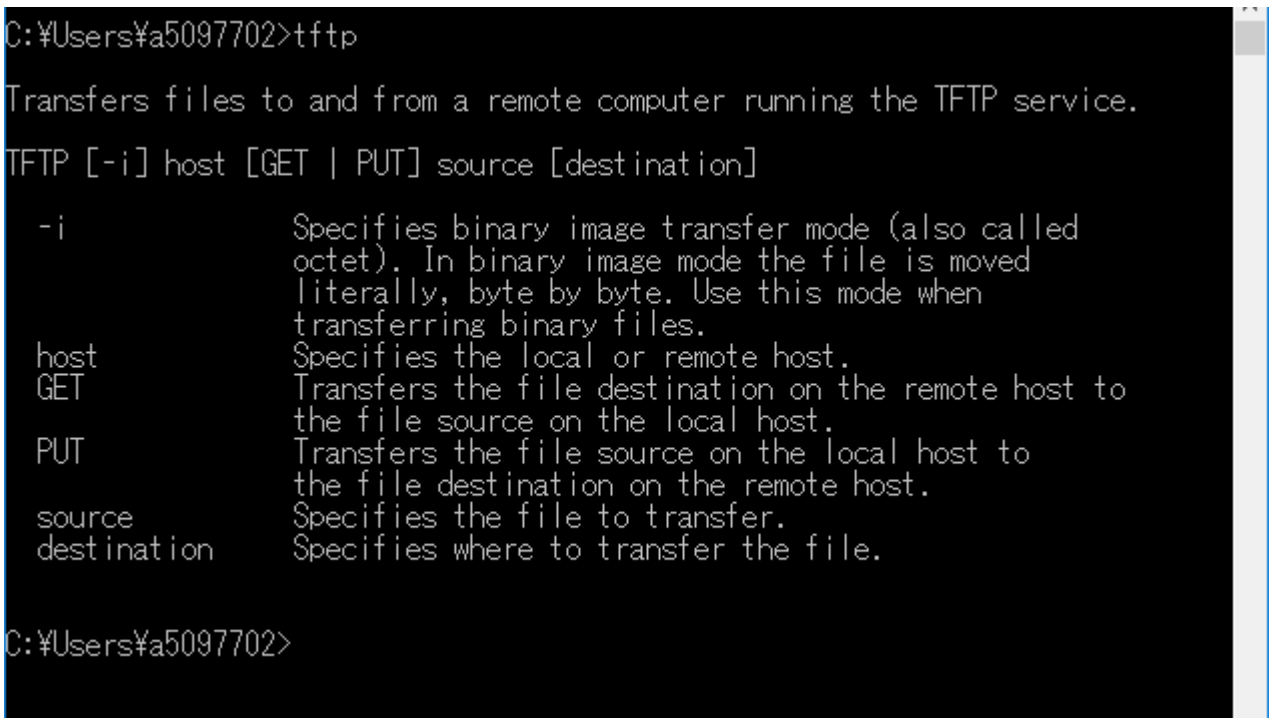
- uNet3_tftp …\Device\Renesas\RIN32M3\Source\Project\uNet3_tftp\IAR\main.eww
- uNet3_tftp_bsd …\Device\Renesas\RIN32M3\Source\Project\uNet3_tftp_bsd\IAR\main.eww

5.2 Confirming Operation

The following describes the procedure for confirming operation by using the standard TFTP client installed on Windows.

• Use the Command Prompt to start the TFTP client.

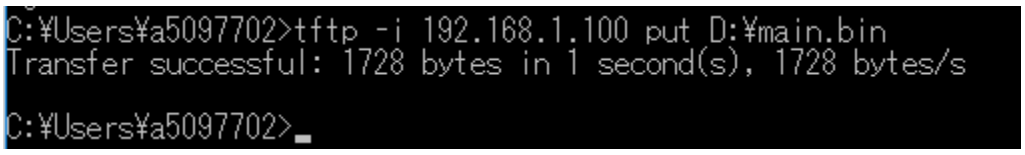
If you input “tftp” with no option, command help is displayed



• The following is an example of transmitting a binary-format file to the TFTP server.

host = 192.168.1.100: TFTP server address

source = main.bin: File for transfer



6. Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	**July. 11, 2017**	**-**	**First version**

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other disputes involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawing, chart, program, algorithm, application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics products.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (space and undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of failure or malfunction of Renesas Electronics products, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.3.0-1 November 2016)

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 6.0