

Renesas Synergy™ Platform

NetX Duo™ FTP Server Module Guide**Introduction**

This module guide will enable you to effectively use a module in your own design. Upon completion of this guide, you will be able to add this module to your own design, configure it correctly for the target application and write code using the included application project code as a reference and efficient starting point. References to more detailed API descriptions and suggestions of other application projects that illustrate more advanced uses of the module are available on the Renesas Synergy™ Knowledge Base as described in the References section at the end of this document, and will be valuable resources for creating more complex designs.

The File Transfer Protocol (FTP) is a protocol designed for file transfers. FTP utilizes reliable Transmission Control Protocol (TCP) services to perform its file transfer function. Because of this, FTP is a highly reliable file transfer protocol. FTP is also high-performance. The actual FTP file transfer is performed on a dedicated FTP connection. NetX™ FTP is specific to IPv4. NetX Duo™ FTP accommodates both IPv4 and IPv6 networks. IPv6 does not directly change the FTP protocol, although some changes in the original NetX FTP API are necessary to accommodate IPv6 and will be described in this document.

This document provides an overview of the key elements related to the NetX Duo FTP Server implementation on the Renesas Synergy Platform. The primary focus is on the addition and configuration of the NetX FTP Server and the NetX Duo FTP Server module to a Renesas Synergy Platform project. For more details on the operation of this module consult the *NetX Duo™ File Transfer Protocol (FTP) User's Guide* for the Renesas Synergy™ Platform document. This document is available as part of an X-Ware™ and NetX™ Component Documents for Renesas Synergy™ zip file from the Renesas Synergy Gallery (<https://synergygallery.renesas.com/ssp/support#read>), and is located at the bottom of the right side list on the web page.

Since NetX FTP Server is nearly identical to NetX Duo FTP Server, this document applies equally to NetX FTP Server, except where noted.

Contents

1. NetX Duo™ FTP Server Module Features.....	3
2. NetX Duo FTP Server Module APIs Overview	3
3. NetX Duo FTP Server Module Operational Overview	4
3.1 NetX Duo FTP Server Module Important Operational Notes and Limitations.....	6
3.1.1 NetX Duo FTP Server Module Operational Notes	6
3.1.2 NetX Duo FTP Server Module Limitations	6
4. Including the NetX Duo FTP Server Module in an Application	7
5. Configuring the NetX Duo FTP Server Module	8
5.1 Configuration Settings for the NetX Duo FTP Server Components.....	9
5.2 NetX Duo FTP Server Module Clock Configuration.....	12
5.3 NetX Duo FTP Server Module Pin Configuration.....	12
6. Using the NetX Duo FTP Server Module in an Application.....	13
7. NetX Duo FTP Server Module Application Project.....	14

- 8. Customizing the NetX Duo FTP Server Module for a Target Application..... 15
- 9. Running the NetX Duo FTP Server Module Application Project 15
- 10. NetX Duo FTP Server Module Conclusion..... 17
- 11. NetX Duo FTP Server Module Next Steps 17
- 12. NetX Duo FTP Server Module Reference Information 18
- Revision History 20

1. NetX Duo™ FTP Server Module Features

- NetX is IPv4 specific
- NetX Duo supports both IPv4 and IPv6 networks
- Works with FileX® file system
- No limit to file name size; uses NULL terminated ASCII strings
- Supports TCP port 21 to field client requests
- Provide high-Level APIs for creating, starting, stopping and deleting service
- NetX FTP and NetX Duo FTP are compliant with RFC 959, RFC 2428 and related RFCs.

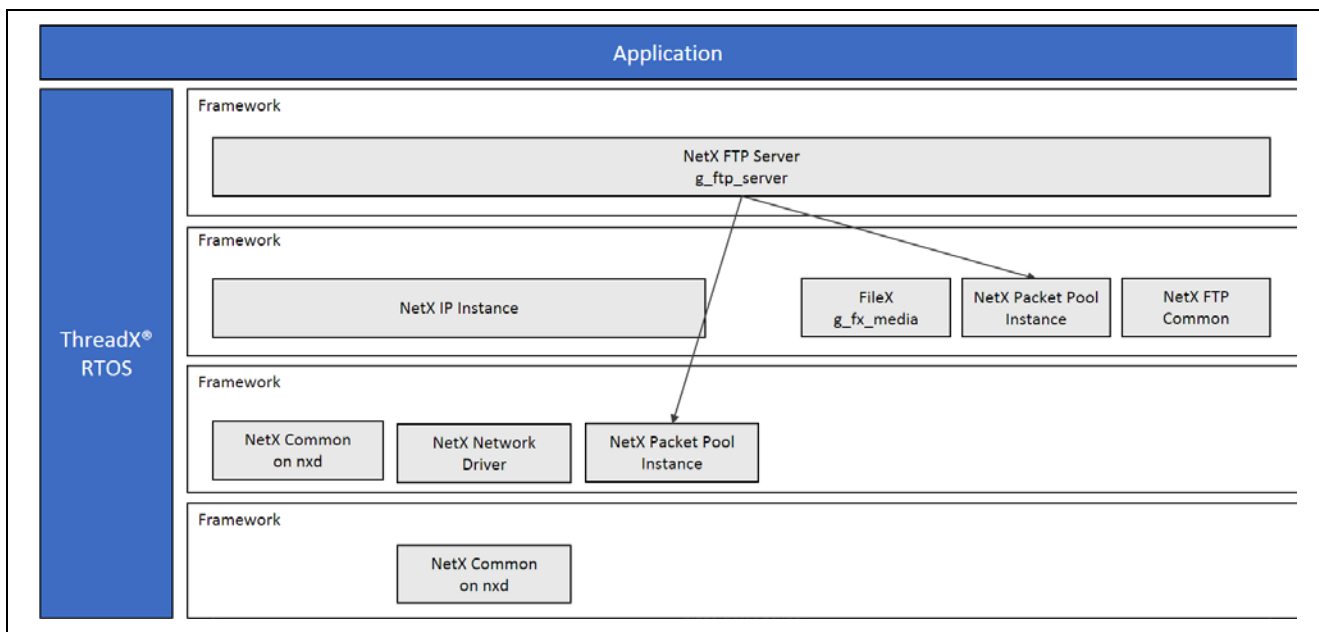


Figure 1. NetX/NetX Duo FTP Server Module Block Diagram

2. NetX Duo FTP Server Module APIs Overview

The NetX Duo FTP Server defines APIs for creating, deleting, starting, and stopping service. A complete list of the available APIs, an example API call and a short description of each can be found in the following table. A table of status return values follows the API summary table.

Table 1. NetX Duo FTP Server Module API Summary

Function Name	Example API Call and Definition
nx_ftp_server_create	<code>nx_ftp_server_create(&my_server, "My Server Name", &ip_0, &ram_disk, stack_ptr, stack_size, &pool_0, my_login, my_logout);</code> Create FTP Server with IPv4 support only.
nxd_ftp_server_create**	<code>nxd_ftp_server_create(&my_server, "My Server Name", &ip_0, &ram_disk, stack_ptr, stack_size, &pool_0, my_duo_login, my_duo_logout);</code> Create FTP Server with IPv4 and IPv6 support.
nx_ftp_server_delete	<code>nx_ftp_server_delete(&my_server);</code> Delete FTP Server.
nx_ftp_server_start	<code>nx_ftp_server_start(&my_server);</code> Start FTP Server.
nx_ftp_server_stop	<code>nx_ftp_server_stop(&my_server);</code> Stop FTP Server.

** Available in NetX Duo FTP Server only

Note: For more complete descriptions of operation and definitions for the function data structures, typedefs, defines, API data, API structures, and function variables, review the *SSP User's Manual API References* for the associated module.

Table 2. Status Return Values

Name	Description
NX_SUCCESS	Successful FTP server function.
NX_PTR_ERROR	Invalid FTP pointer.
NX_CALLER_ERROR	Invalid caller of this service.

Note: Lower-level drivers may return common error codes. Refer to the *SSP User's Manual*, API References for the associated module for a definition of all relevant status return values.

3. NetX Duo FTP Server Module Operational Overview

Because FTP Client and Server operations are so closely linked, the following descriptions cover key elements of both Client and Server operations.

FTP Requirements

The NetX Duo FTP package requires NetX Duo for proper operation and a previously created IP instance for running NetX Duo services and periodic tasks. If running the FTP host application over an IPv6 network, IPv6, and ICMPv6 are automatically enabled on the IP instance. TCP is also enabled by default for either IPv6 or IPv4 networks. The IPv6 host application must set its link local and global IPv6 address either directly or in the case of global address through DHCPv6.

The FTP Server is also designed to work with the FileX embedded file system. If FileX is not available, the host developer can implement or substitute their own file system along the guidelines suggested in `filex_stub.h` by defining each of the services listed in that file. This is discussed in later sections of this guide.

The FTP Server portion of the NetX Duo FTP package has several additional requirements. It requires complete access to TCP well-known port 21 for handling all Client FTP command requests and well-known port 20 for handling all Client FTP data transfers.

FTP File Names

FTP file names should be in the format of the target file system, usually FileX. They should be NULL terminated ASCII strings, with full path information if necessary. There is no specified limit for the size of FTP file names in the NetX Duo FTP implementation. The packet pool payload size should be able to accommodate the maximum path and/or file name.

Table 3. FTP Commands

FTP Command	Meaning
CWD path	Change working directory
DELE filename	Delete specified file name
EPRT ip_address, port	Provide IPv6 address and Client data port
LIST directory	Get directory listing
MKD directory	Make new directory
NLST directory	Get directory listing
NOOP	No operation, returns success
PASS password	Provide password for login
PORT ip_address, port	Provide IP address and Client data port
PWD path	Pickup current directory path
QUIT	Terminate Client connection
RETR filename	Read specified file
RMD directory	Delete specified directory
RNFR oldfilename	Specify file to rename
RNTO newfilename	Rename file to supplied file name
STOR filename	Write specified file
TYPE I	Select binary file image
USER username	Provide username for login

FTP Server Responses

Once the FTP Server processes the Client request, it returns a 3-digit coded response in ASCII followed by optional ASCII text. The numeric response is used by the FTP Client software to determine whether the operation succeeded or failed. The following list shows various FTP Server responses to Client requests:

Table 4. First Numeric Field

First Numeric Field	Meaning
1xx	Positive preliminary status – another reply coming.
2xx	Positive completion status.
3xx	Positive preliminary status – another command must be sent.
4xx	Temporary error condition.
5xx	Error condition.

Table 5. Second Numeric Field

Second Numeric Field	Meaning
x0x	Syntax error in command.
x1x	Informational message.
x2x	Connection related.
x3x	Authentication related.
x4x	Unspecified.
x5x	File system related.

For example, a Client request to disconnect a FTP connection with the QUIT command will typically be responded with a **221** code from the Server – if the disconnect is successful.

FTP Communication

The FTP Server utilizes the well-known TCP port 21 to field Client requests. The general sequence of FTP events are as follows. As mentioned previously, the only difference with FTP running over IPv6 is the PORT command is replaced with the EPRT command.

FTP Read File Requests

1. Client issues TCP connect to Server port 21.
2. Server sends **220** response to signal success.
3. Client sends **USER** message with "username."
4. Server sends **331** response to signal success.
5. Client sends **PASS** message with "password."
6. Server sends **230** response to signal success.
7. Client sends **TYPE I** message for binary transfer.
8. Server sends **200** response to signal success.
9. Client sends **PORT** message with IP address and port.
10. Server sends **200** response to signal success.
11. Client sends RETR message with file name to read.
12. Server creates data socket and connects with client data port specified in the **EPRT** command.
13. Server sends **125** response to signal file read has started.
14. Server sends contents of file through the data connection. This process continues until file is completely transferred.
15. When finished, Server disconnects data connection.
16. Server sends **250** response to signal file read is successful.
17. Clients sends **QUIT** to terminate FTP connection.
18. Server sends **221** response to signal disconnect is successful.
19. Server disconnects FTP connection.

FTP Write Requests

1. Client issues TCP connect to Server port 21.
2. Server sends **220** response to signal success.

3. Client sends **USER** message with “username.”
4. Server sends **331** response to signal success.
5. Client sends **PASS** message with “password.”
6. Server sends **230** response to signal success.
7. Client sends **TYPE I** message for binary transfer.
8. Server sends **200** response to signal success.
9. IPv6 applications: Client sends **EPRT** message with IP address and port. IPv4 applications: Client sends **PORT** message with IP address and port.
10. Server sends **200** response to signal success.
11. Client sends **STOR** message with file name to write.
12. Server creates data socket and connects with client data port specified in the previous **EPRT** or **PORT** command.
13. Server sends **125** response to signal file write has started.
14. Client sends contents of file through the data connection. This process continues until file is completely transferred.
15. When finished, Client disconnects data connection.
16. Server sends **250** response to signal file write is successful.
17. Client sends **QUIT** to terminate FTP connection.
18. Server sends **221** response to signal disconnect is successful.
19. Server disconnects FTP connection.

FTP Authentication

Whenever a FTP connection takes place, the Server receives a username and password from the Client. Some FTP sites allow what is called Anonymous FTP, which allows FTP access without a specific username and password. For this type of connection, **anonymous** should be supplied for username and the password should be a complete e-mail address.

Your application is responsible for supplying NetX Duo FTP with login and logout authentication routines. These are supplied during the `nxd_ftp_server_create` and `nx_ftp_server_create` services and called from the password processing. The difference between the two is the `nxd_ftp_server_create` input function pointers to login and logout authenticate functions expect the NetX Duo address type `NXD_ADDRESS`. This data type holds both IPv4 or IPv6 address formats, making this function the **duo** service supporting both IPv4 and IPv6 networks. The `nx_ftp_server_create` input function pointers to login and logout authenticate functions expect `ULONG` IP address type. This function is limited to IPv4 networks. You are encouraged to use the **duo** service whenever possible.

If the login function returns `NX_SUCCESS`, the connection is authenticated and FTP operations are allowed. Otherwise, if the login function returns something other than `NX_SUCCESS`, the connection attempt is rejected.

3.1 NetX Duo FTP Server Module Important Operational Notes and Limitations

3.1.1 NetX Duo FTP Server Module Operational Notes

FTP RFCs

NetX Duo FTP Server is compliant with RFC 959, RFC 2428 and related RFCs.

3.1.2 NetX Duo FTP Server Module Limitations

The FTP standard has many options regarding the representation of file data. NetX Duo FTP does not implement switch options such as `ls -al`. NetX FTP Server expects to receive requests and their arguments in a single packet rather than consecutive packets.

Like UNIX implementations, NetX FTP Server assumes the following file format constraints:

- File Type: Binary
- File Format: Nonprint Only
- File Structure: File Structure Only

Refer to the most recent *SSP Release Notes* for any additional operational limitations for this module.

4. Including the NetX Duo FTP Server Module in an Application

This section describes how to include the NetX Duo FTP Server module in an application using the SSP configurator.

Note: This section assumes you are familiar with creating a project, adding threads, adding a stack to a thread and configuring a block within the stack. If you are unfamiliar with any of these items, refer to the first few chapters of the *SSP User's Manual* to learn how to manage each of these important steps in creating SSP-based applications.

To add NetX Duo FTP Server to an application, simply add a thread in the Configurator Panel. Then using the thread stacks, choose **X-Ware > NetX Duo (or NetX) > FTP Server** as shown in the following table. The default name for the NetX Duo FTP Server is `g_ftp_server0`. This name can be changed in the associated Properties window.

Table 6. FTP Server Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_ftp_server0 NetX FTP Server	Threads	New Stack> X-Ware> NetX> Protocols> NetX FTP Server
g_ftp_server0 NetX Duo FTP Server	Threads	New Stack> X-Ware> NetX Duo> Protocols> NetX FTP Server

When the NetX Duo FTP Server is added to the thread stack as shown in the following figure, the configurator automatically adds the needed lower-level drivers. Any drivers that need additional configuration information will be box text highlighted in red. Modules with a gray band are individual modules that stand alone. Modules with a blue band are shared or common and need only be added once, since they can be used by multiple stacks. Modules with a pink band can require the selection of lower level drivers. Sometimes these are optional or recommended and this is indicated in the block with the inclusion of this text. If the addition of lower level drivers is required, the module description will include "Add" in the text. Clicking on any pink banded modules will bring up the "New" icon and then will show the possible choices.

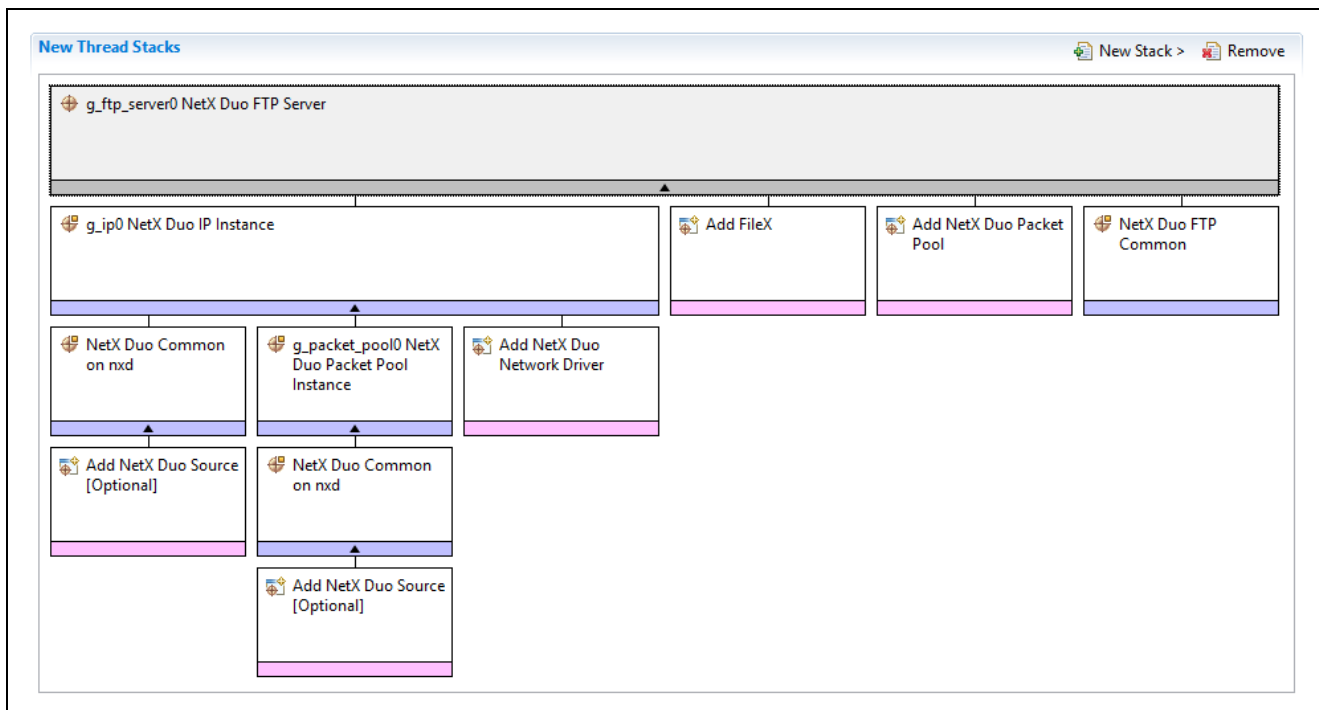


Figure 2. NetX/NetX Duo FTP Server Module Stack

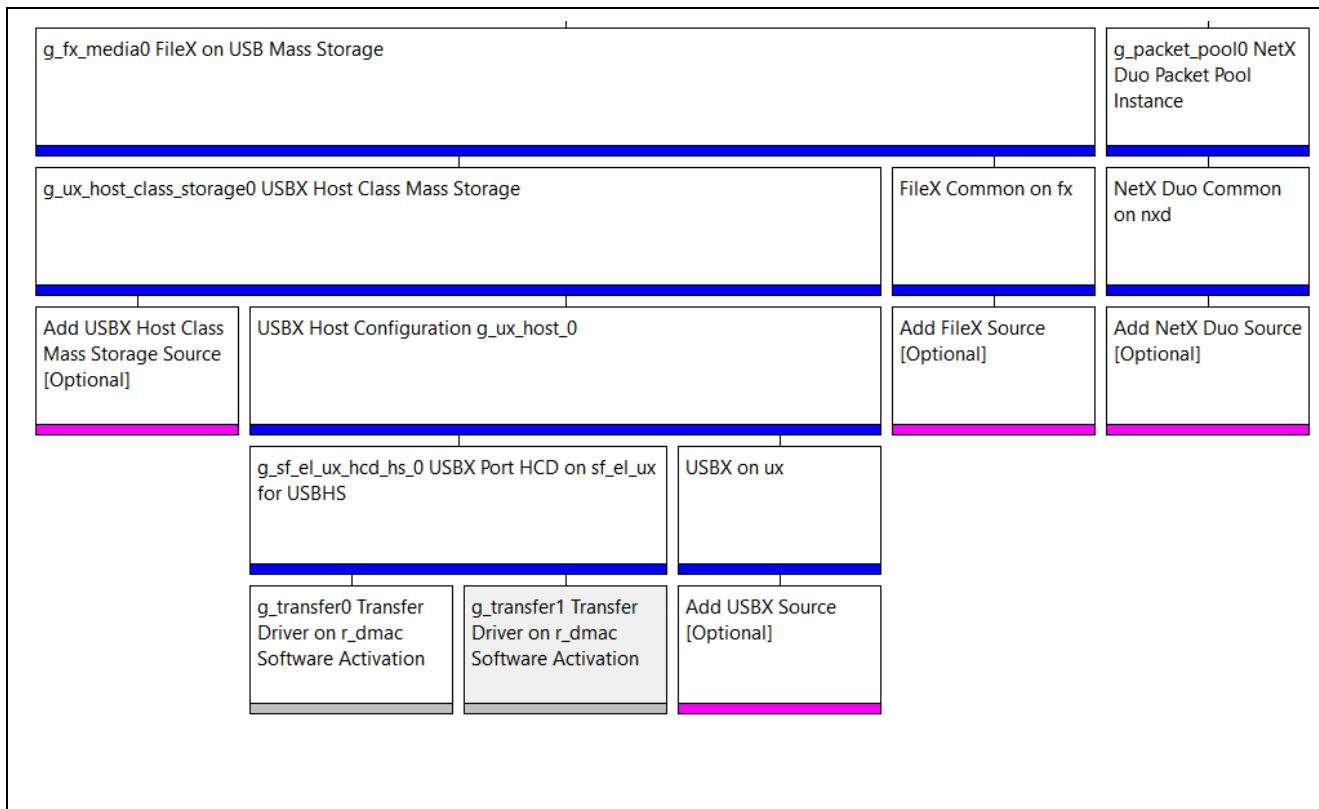


Figure 3. FileX on Mass Storage Module Stack

For the FileX on Mass Storage, select the Add FileX box and click on New -> FileX on USB Mass Storage. Then select the Add USBX Port HCD and choose New -> USBX Port HCD on sf_el_ux for USBHS. Finally select the Add Transfer Module for TX and choose the r_dmac option. Repeat for the Transfer Module for RX. At this point, the FileX on Mass Storage should resemble the figure above.

5. Configuring the NetX Duo FTP Server Module

The NetX FTP Server module must be configured by you for the desired operation. The SSP configuration window will automatically identify by highlighting the block in red, and any required configuration selections, such as interrupts or operating modes, which must be configured for lower-level modules for successful operation. Only those properties that can be changed without causing conflicts are available for modification. Other properties are 'locked' and not available for changes and are identified with a lock icon for the 'locked' property in the Properties window in the ISDE. This approach simplifies the configuration process and makes it much less error-prone than previous 'manual' approaches to configuration. The available configuration settings and defaults for all the user-accessible properties are given in the Properties tab within the SSP configurator and are shown in the following tables for easy reference.

One of the properties most often identified as requiring a change is the interrupt priority; this configuration setting is available within the Properties window of the associated module. Simply select the indicated module and then view the Properties window. The interrupt settings are often toward the bottom of the properties list, so scroll down until they become available. Note that the interrupt priorities listed in the Properties window in the ISDE will include an indication as to the validity of the setting based on the MCU targeted (CM4 or CM0+). This level of detail is not included in the following configuration properties tables, but is easily visible with the ISDE when configuring interrupt-priority levels.

Note: You may want to open your ISDE, create the module and explore the property settings in parallel with looking over the following configuration table settings. This will help orient you and can be a useful 'hands-on' approach to learning the ins and outs of developing with SSP.

Table 7. Configuration Settings for the NetX FTP Server and the NetX Duo FTP Server Module

Parameter	Value	Description
Internal Thread Priority	16	Internal thread priority selection
Maximum clients to serve simultaneously	4	Maximum number of clients allowed

Parameter	Value	Description
Duration internal services will suspend for (seconds)	100	Duration internal services will suspend for selection
Maximum username length (bytes)	20	Maximum username length selection
Maximum password length (bytes)	20	Maximum password length selection
Duration allowed with no activity (seconds)	240	Duration allowed with no activity
Duration for initial timeout (seconds)	2	Duration for initial timeout
Maximum queued transmit packets	20	Maximum queued transmit selection
Maximum retries per packet	10	Maximum retries per selection
Binary left shift as multiplier for retry duration	1	Binary left shift as multiplier for retry duration
Name	g_ftp_server0	
Internal thread stack size (bytes)	4096	Internal thread stack size; smaller stack size may lead to stack overflow
Name of Login Function	ftp_login	Name of Login selection
Name of Logout Function	ftp_logout	Name of Logout selection

Note: The example values and defaults are for a project using the S7G2 Synergy MCU. Other MCUs may have different default values and available configuration settings.

In some cases, settings other than the defaults for lower level modules can be desirable. For example, it might be useful to select different MAC Addresses. The configurable properties for the lower level stack modules are given in the following sections for completeness and as a reference.

Note: Most of the property settings for lower-level modules are intuitive and usually can be determined by inspection of the associated Properties window with the SSP configurator.

5.1 Configuration Settings for the NetX Duo FTP Server Components

Only a small number of settings must be modified from the default for lower-level drivers as indicated by the red text in the thread stack block. Notice that some of the configuration properties must be set to a certain value for proper framework operation and will be locked to prevent user modification. The following table identifies all the settings within the properties section for the module.

Note: The example values and defaults are for a project using the S7G2 Synergy MCU. Other MCUs may have different default values and available configuration settings.

Rows in yellow must be defined for your application environment. Rows in green have recommended values that differ from the default value:

Table 8. Configuration Settings for the NetX IP Duo Instance

ISDE Property	Value	Description
Name	g_ip0	Module name
IPv4 Address (use commas for separation)	192,168,0,2	IPv4 Address selection
Subnet Mask (use commas for separation)	255,255,255,0	Subnet Mask selection
IP Helper Thread Stack Size (bytes)	Recommended 2048 Default 1024	IP Helper Thread Stack Size (bytes) selection
IP Helper Thread Priority	Recommended 1 Default 3	IP Helper Thread Priority selection. Should be the highest priority of all threads using NetX services.
ARP	Enable	ARP selection
ARP Cache Size in Bytes	520	ARP Cache Size in Bytes selection
Reverse ARP	Enable, Disable (Default: Enable)	Reverse ARP selection
TCP	Enable	TCP selection

ISDE Property	Value	Description
UDP	Enable, Disable (Default: Enable)	UDP selection
ICMP	Enable, Disable (Default: Enable)	ICMP selection
IGMP	Enable, Disable (Default: Enable)	IGMP selection

Table 9. Configuration Settings for the NetX Duo Packet Pool Instance

ISDE Property	Value	Description
Name	g_packet_pool0	Module name
Packet Size in Bytes	Recommended 1568 Default 640	Packet size selection. Setting to 1568 eliminates the requirement for packet chaining
Number of Packets in Pool	16	Number of packets in pool selection. May require a higher number in bursty or heavy network traffic conditions.

Table 10. Configuration Settings for USBx on ux

ISDE Property	Value	Description
USBX Pool Memory Name	g_ux_pool_memory	Module name
USBX Pool Memory Size	Recommended 63488 Default 18356	USBX memory pool size
User Callback for Host Event Notification (Hosts only)	usb_host_plug_event_notification	Default is NULL which will not work with this project.

Table 11. Configuration Settings for the FileX on Block Media (not used in this project)

ISDE Property	Value	Description
Name	g_fx_media0	Module name.
Format media during initialization	Enabled, Disabled (Default: Disabled)	Format media during initialization selection.
File System is on SDMMC	True, False (Default: True)	File System initialization selection.
Directory Entries	256	Directory entries selection.
Hidden Sectors	0	Hidden sectors selection.
Total Sectors	3751936	Total sectors selection.
Bytes per Sector	512	Bytes per Sector selection.
Sectors per Cluster	1	Sectors per Cluster selection.
Working media memory size	512	Working media memory size selection.

Table 12. Configuration for the FileX on USB Mass Storage

ISDE Property	Setting	Description
Name	g_fx_media0	Module name
Name of FileX Media Control block Initialization	fx_media_init_function	Function initializing FileX Media in auto-generated code.
Auto Media Initialization	Disable	Determines if above function is called
Timeout ticks for Media Initialization	1000	Timeout for media initialization function to complete

Table 13. Configuration for the USBX PORT HCD for g_sf_el_ux_hcd_hs_0

ISDE Property	Setting	Description
Name	g_sf_el_ux_hcd_hs_0	Module name
USB Controller Selection	USBHS	USB port type: High speed vs Full Speed
High Speed Interrupt Priority	Priority 3	Priority of High Speed interrupt (default is disabled)
FIFO size for Bulk Pipes	512 bytes	Transfer size per pipe
VBUSEN pin Signal Logic	Active Low	Pin setting for HS port
Enable High Speed	Enable	Enable high speed transfers

Table 14. Configuration for Transfer Driver RX on r_dmac g_transfer0

ISDE Property	Setting	Description
Parameter Checking	BSP	Enable or disable the parameter checking
Name	g_transfer0	Channel 0 Phy reset pin selection
Channel	0	Channel
Interrupt Priority	Priority 4	Enable interrupts (default is disabled)

Table 15. Configuration for Transfer Driver TX on r_dmac g_transfer1

ISDE Property	Setting	Description
Parameter Checking	BSP	Enable or disable the parameter checking
Name	g_transfer0	Channel 0 Phy reset pin selection
Channel	1	Channel
Interrupt Priority	Priority 4	Enable interrupts (default is disabled)

Table 16. Configuration for NetX Duo FTP Common

ISDE Property	Setting	Description
FileX support	Enabled, Disabled (Default: Enabled)	FileX support selection
Control Type of Service	Normal, Minimum delay, Maximum data, Maximum reliability, Minimum cost (Default: Normal)	Control type of service selection
Data Type of Service	Normal, Minimum delay, Maximum data, Maximum reliability, Minimum cost (Default: Normal)	Data type of service selection
Fragmentation option	Don't fragment, Fragment okay (Default: Don't fragment)	Fragment option selection
Time to live	128	Time to live selection
Packet Queue depth	60	Packet queue depth selection

Table 17. Configuration for NetX Duo Common on nxd

ISDE Property	Setting	Description
No configurable settings		

Note: The example values and defaults are for a project using the S7G2 Synergy MCU. Other MCUs may have different default values and available configuration settings.

Table 18. Configuration for NetX Duo Port Ether

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled (Default: BSP)	Enable or disable the parameter checking
Channel 0 Phy Reset Pin	IOPORT_PORT_09_PIN_03	Channel 0 Phy reset pin selection
Channel 0 MAC Address High Bits	0x00002E09	Channel 0 MAC address high bits selection
Channel 0 MAC Address Low Bits	0x0A0076C7	Channel 0 MAC address low bits selection
Channel 1 Phy Reset Pin	IOPORT_PORT_08_PIN_06	Channel 1 Phy reset pin selection
Channel 1 MAC Address High Bits	0x00002E09	Channel 1 MAC address high bits selection
Channel 1 MAC Address Low Bits	0x0A0076C8	Channel 1 MAC address low bits selection
Number of Receive Buffer Descriptors	8	Number of receive buffer descriptors selection
Number of Transmit Buffer Descriptors	32	Number of transmit buffer descriptors selection
Ethernet Interrupt Priority	Default: Disabled Priority 3-15	Ethernet interrupt priority selection
Name	g_sf_el_nx	Module name
Channel	1	Channel selection
Callback	NULL	Callback selection

5.2 NetX Duo FTP Server Module Clock Configuration

The ETHERC peripheral module uses PCLKA as its clock source. The PCLKA frequency is set by using the SSP configurator clock tab, prior to a build, or by using the CGC Interface at run-time.

5.3 NetX Duo FTP Server Module Pin Configuration

The ETHERC peripheral module uses pins on the MCU to communicate to external devices. I/O pins must be selected and configured as required by the external device. The following table illustrates the method for selecting the pins within the SSP configuration window and the subsequent table illustrates an example selection for the ETHERC pins.

Note: For some peripherals, the operation-mode selection determines what peripheral signals are available and what MCU pins are required.

Table 19. Pin Selection Sequence for SCI1

Resource	ISDE Tab	Pin Selection Sequence
ETHERC	Pins	Select Peripherals > Connectivity:ETHERC >ETHERC1.RMII

Note: The selection sequence assumes SCI1 is the desired hardware target for the driver.

Table 20. Pin Configuration Settings for the NetX Duo FTP Server Module on SCI

Pin Configuration Property	Value	Description
Operation Mode	Disabled, Custom, RMII (Default: Disabled)	Select RMII as the Operation Mode for ETHERC1
Pin Group Selection	Mixed, _A only (Default: _A only)	Pin group selection
REF50CK	P701	REF50CK Pin
TXD0	P700	TXD0 Pin
TXD1	P406	TXD1 Pin
TXD_EN	P405	TXD_EN Pin
RXD0	P702	RXD0 Pin
RXD1	P703	RXD1 Pin
RX_ER	P704	RX_ER Pin
CRS_DV	P705	CRS_DV Pin
MDC	P403	MDC Pin
MDIO	P404	MDIO Pin

Table 21. Pin Configuration Settings for the NetX Duo FTP Server Module on SCI

Pin Configuration Property	Value	Description
Operation Mode	Disabled, Custom, RMII (Default: Disabled)	Select RMII as the Operation Mode for ETHERC1
Pin Group Selection	Mixed, _A only (Default: _A only)	Pin group selection
REF50CK	P701	REF50CK Pin
TXD0	P700	TXD0 Pin
TXD1	P406	TXD1 Pin
TXD_EN	P405	TXD_EN Pin
RXD0	P702	RXD0 Pin
RXD1	P703	RXD1 Pin
RX_ER	P704	RX_ER Pin
CRS_DV	P705	CRS_DV Pin
MDC	P403	MDC Pin
MDIO	P404	MDIO Pin

6. Using the NetX Duo FTP Server Module in an Application

The following example assumes an IP Instance for the FTP server with ARP has already been created and TCP enabled. The typical steps in using the NetX FTP server in an application are:

1. Wait for the network to be enabled.
2. Define login and logout callback functions (optional)
3. Start the previously created FTP Server using the nx_ftp_server_start API
4. At this point the FTP server simply accepts FTP Client requests and transfers files and processes other FTP commands as requested until nx_ftp_server_stop is called.

5. These common steps are illustrated in a typical operational flow diagram in the following figure:

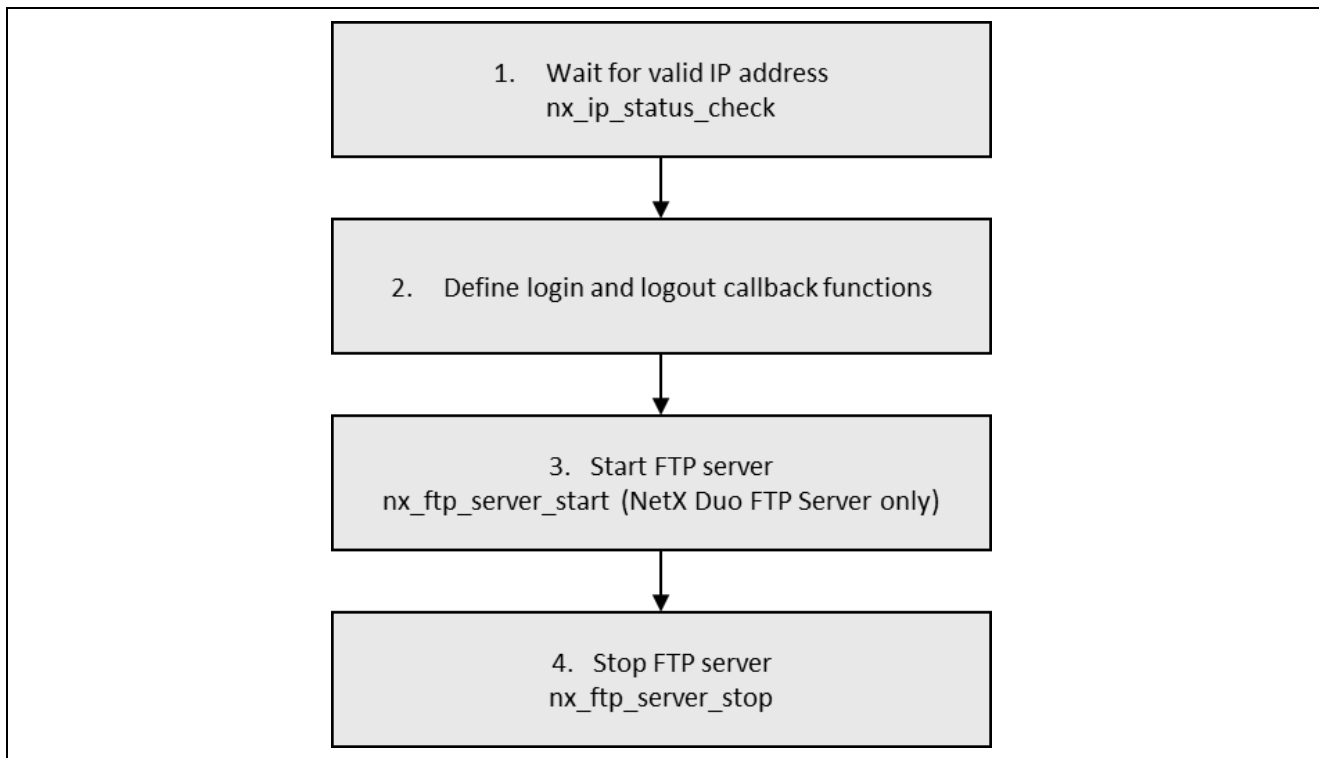


Figure 4. Flow Diagram of a Typical NetX Duo FTP Server Module Application

7. NetX Duo FTP Server Module Application Project

The Application Project associated with this module guide demonstrates the steps in a full design. The project can be found as described in the References Section at the end of this document. You may want to import and open the application project within the ISDE and view the configuration settings for the FTP Server module. You can also read over the code (in `ftp_server_thread_entry.c`) used to illustrate the FTP Server APIs in a complete design.

The application project demonstrates the typical use of the FTP Server APIs. The application project `ftp_server_thread_entry` waits for the network to be enabled. It then waits until USB storage device is attached to the USB port. Lastly, the FTP server is initialized. The application then starts the FTP server thread by calling `nx_ftp_server_start` and lets it process Client connection requests and FTP commands.

LED1 is used in the application as a sanity check and is blinked by an independent blinky thread. The main loop on the FTP Server thread blinks LED3 to indicate activity. The LED2 (red) is used as an error indicator; if it lights up, an error has occurred.

Table 22. Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e ² studio	v7.3.0	Integrated Solution Development Environment
SSP	v1.6.0	Synergy Software Platform
SK-S7G2	v3.0 or later	Development Kit
IAR EW for Renesas Synergy	v8.23.3	IAR Embedded Workbench for Renesas Synergy
SSC	v7.3.0	Synergy Standalone Configurator

8. Customizing the NetX Duo FTP Server Module for a Target Application

Some configuration settings will normally be changed by the developer from those shown in the Application Project. For example, you can easily change the configuration settings for the FTP Server in the configurator Properties window.

The FTP Server can use any supported NetX interface. Consult the *WiFi Framework* documentation to see how to configure a NetX Duo stack to work over WiFi.

The FTP Server can use any `FX_MEDIA`, either physical or virtual. FileX can be configured to use different underlying media or even a RAM disk. Look at the FileX documentation for examples.

FileX can be replaced by a proprietary file system as long as all services specified in `fileX_stub.h` are defined.

9. Running the NetX Duo FTP Server Module Application Project

To run the FTP Server module application project and to see it executed on a target kit, you can simply import it into your ISDE, compile and run debug. Refer to the *Synergy Project Import Guide* (r11an0023eu0121-synergy-ssp-import-guide.pdf, included in this package) for instructions on importing the project into e² studio or IAR Embedded Workbench for Synergy™ and build/run the application.

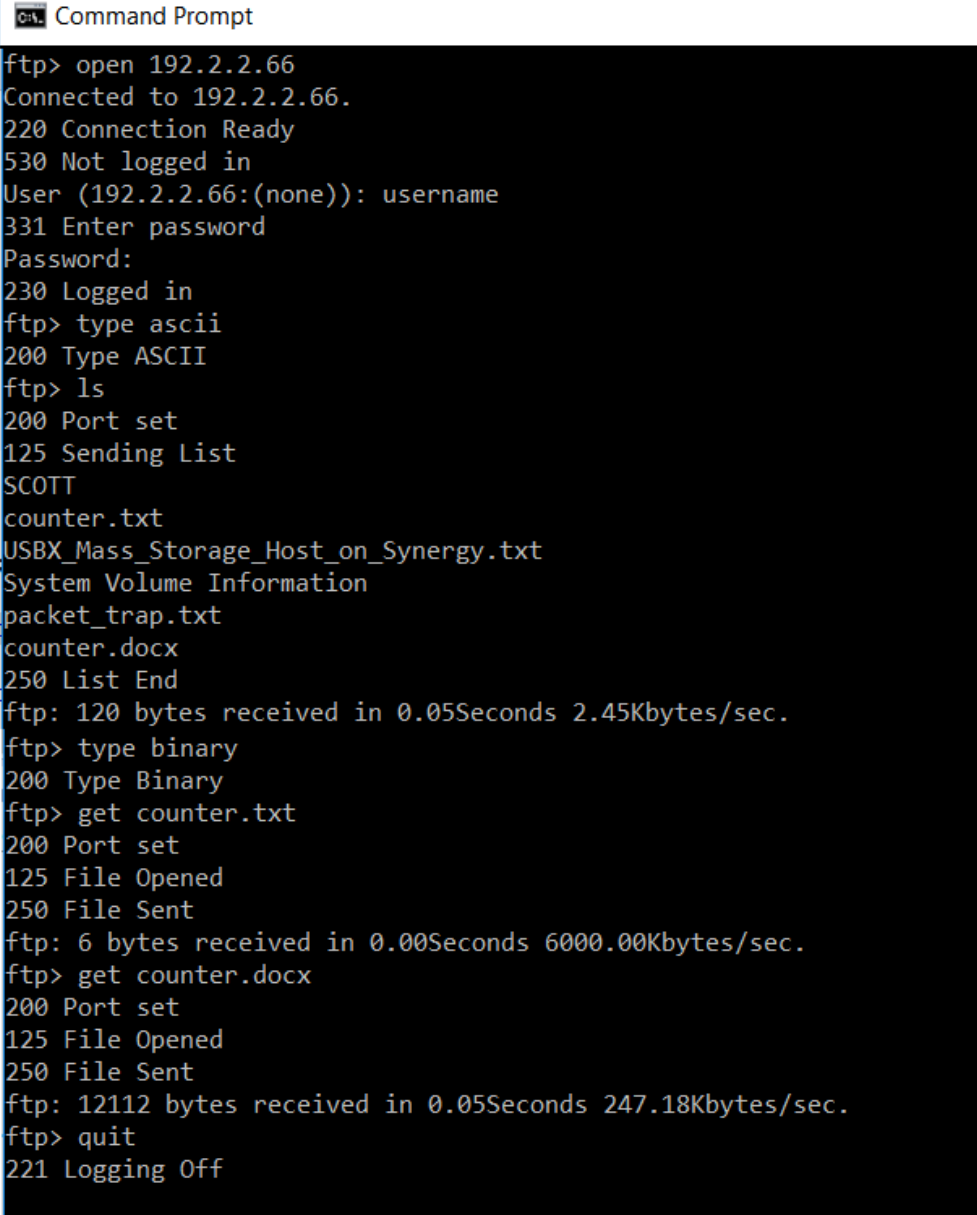
To implement the FTP Server module application in a new project, follow the steps below for defining, configuring, auto-generating files, adding code, compiling, and debugging on the target kit. Following these steps in a hands-on approach that can help make the development process with SSP more practical.

Note: The following steps are described in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are not familiar, refer to the first few chapters of the *SSP User's Manual* for a description of how to accomplish these steps.

To create and run the FTP Server application project, simply follow these steps:

1. Refer to the Synergy Project Import Guide (r11an0023eu0121-synergy-ssp-import-guide.pdf, included in this package) for instructions on importing the project into e² studio or IAR embedded workbench and building/running the application
2. Connect to the host PC via a micro USB cable to J19 on SK-S7G2.
3. Connect the SK-S7G2 board with an Ethernet cable to a local network on which the FTP Client resides.
Note: Please make sure the server the client resides on the same subnet to run this application.
4. Plug a USB mass storage device to the USB_H port of the SK-S7G2 board. Make sure that the device has been formatted in one for the supported file systems and contains some files to be served.
5. Start to debug the application
6. Debug output can be viewed in the Renesas Virtual Debug Console
7. Start an FTP Client session on a PC on the local network, using the built in utility on windows or Linux, or a similar utility:
 - Open a connection with the FTP server
 - Log in with username and password
 - Type ASCII for ascii transfer
 - Send the ls command for what files exist on the FTP server (mass storage device)
 - Either get a file existing on the Server or put a file from the local network to the FTP server
 - Log out

Following is an example FTP Server session in which an FTP client on a PC connects to the FTP server on the SK-S7G2 board from a command shell. After making the TCP connection, the FTP Client supplies a username and password and is logged in. It asks for a list of all files on the FileX mass storage using the ls command. Then it specifies the binary mode by typing “type binary” and gets the counter.docx and counter.txt files. It closes the session with the quit command.



```
Command Prompt
ftp> open 192.2.2.66
Connected to 192.2.2.66.
220 Connection Ready
530 Not logged in
User (192.2.2.66:(none)): username
331 Enter password
Password:
230 Logged in
ftp> type ascii
200 Type ASCII
ftp> ls
200 Port set
125 Sending List
SCOTT
counter.txt
USBX_Mass_Storage_Host_on_Synergy.txt
System Volume Information
packet_trap.txt
counter.docx
250 List End
ftp: 120 bytes received in 0.05Seconds 2.45Kbytes/sec.
ftp> type binary
200 Type Binary
ftp> get counter.txt
200 Port set
125 File Opened
250 File Sent
ftp: 6 bytes received in 0.00Seconds 6000.00Kbytes/sec.
ftp> get counter.docx
200 Port set
125 File Opened
250 File Sent
ftp: 12112 bytes received in 0.05Seconds 247.18Kbytes/sec.
ftp> quit
221 Logging Off
```

Figure 5. Command Shell of FTP Server Client Session

The corresponding output in Renesas Virtual Debug Console is as shown in the following graphic.

The image shows a screenshot of the Renesas Virtual Debug Console. At the top, there are tabs for 'Console', 'Tasks', 'Renesas Coverage', 'Memory Usage', and 'Performance'. The 'Console' tab is active, displaying the following text:

```
Renesas Debug Virtual Console
Waiting for a USB storage device to become ready...
Waiting for a USB storage device insertion...
Waiting for a USB storage device insertion...
Waiting for a USB storage device insertion...
Waiting for a USB storage device insertion...
Waiting for a USB storage device insertion...
Storage device ready.

Connection requests 1
total_bytes_sent and received 0, 0
allocation_errors 0
relisten_errors 0

Connection requests 1
total_bytes_sent and received 0, 0
allocation_errors 0
relisten_errors 0

Connection requests 1
total_bytes_sent and received 12112, 0
allocation_errors 0
relisten_errors 0

Connection requests 1
total_bytes_sent and received 24224, 0
allocation_errors 0
relisten_errors 0

Connection requests 1
total_bytes_sent and received 36336, 0
allocation_errors 0
relisten_errors 0
```

Figure 6. Renesas Virtual Debug Console Output

10. NetX Duo FTP Server Module Conclusion

This module guide has provided all the background information needed to select, add, configure and use the module in an example project. Many of these steps were time consuming and error-prone activities in previous generations of embedded systems. The Renesas Synergy Platform makes these steps much less time consuming and removes the common errors, like conflicting configuration settings or the incorrect selection of low level modules. The use of high level APIs as demonstrated in this application project, illustrates additional development-time savings by allowing work to begin at a high level and avoiding the time required in older development environments to use, or in some cases, create, lower-level drivers.

An FTP Server allows you to provide access to file data from a variety of different clients, as the File Transfer Protocol is widely supported. As seen in this note, the use of the FTP Server module requires only a few steps and provides full capabilities.

11. NetX Duo FTP Server Module Next Steps

After you have mastered a simple FTP Server project you may want to review a more complex example.

The FTP Server login and logout are only placeholders with no real implementation. You may want to implement them to verify access to the FTP Server.

You may also want to replace the USB storage device interface by a custom wrapper and serve virtual data generated by your application instead of files coming from a real file system.

Other Application Projects and Application Notes that demonstrate FTP Server use can be found as described in the next section.

12. NetX Duo FTP Server Module Reference Information

SSP User's Manual: Available in html format in the SSP distribution package and as a pdf from the Synergy Gallery.

Links to all the most up-to-date `ftp_server` module reference materials and resources are available on the Synergy Knowledge Base: https://en-us.knowledgebase.renesas.com/English_Content/Renesas_Synergy%E2%84%A2_Platform/Renesas_Synergy_Knowledge_Base/FTP_Server_Module_Guide_References.

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan.11.18	-	Initial release
1.01	Apr.30.19	-	Updated for SSP v1.6.0

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 claims 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, drawings, charts, programs, algorithms, and 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 reverse engineer 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, copying or reverse engineering.
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; 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.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; 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 any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using 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 dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of 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. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a 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. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using 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 comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
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.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.