

Introduction

This Module Guide will enable you to effectively use this module in your own design. On completion of this guide you will be able to configure it correctly for the target application and write code, using the included Application Project code as a reference and an efficient starting point. References to more detailed API descriptions and suggestions of other Application Projects that illustrate more advanced uses of the module are included in the 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 with high-performance. The actual FTP file transfer is performed on a dedicated FTP connection.

The NetX™ FTP Client supports IPv4 only. The NetX Duo™ FTP Client 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 FTP Client and the NetX Duo FTP Client implementation on the Renesas Synergy™ Platform. The primary focus is on the addition and configuration of the NetX FTP Client and the NetX Duo FTP Client module to a Renesas Synergy Platform project. For more details on the operation of this module, consult the *NetX Duo File Transfer Protocol (FTP) Client 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.

This document is divided into the following sections that can be read independently by an experienced Synergy developer or in series by developers new to the Renesas Synergy Platform.

Contents

1. NetX and NetX Duo FTP Client Features.....	3
2. NetX and NetX Duo FTP Client APIs Overview	3
3. NetX and NetX Duo FTP Client Operational Overview.....	5
3.1 Important Operational Notes and Limitations	7
3.1.1 Operational Notes	7
3.1.2 Limitations	7
4. Including the NetX FTP and NetX Duo FTP Client in an Application	8
5. Configuring the NetX and NetX Duo FTP Client Module	8
5.1 Configuration Settings for the NetX and NetX Duo FTP Client Low Level Drivers	9
5.2 Clock Configuration	12
5.3 Pin Configuration	12
6. Using the NetX FTP Client in an Application	13
7. The NetX FTP Client Application Project	14
8. Customizing the NetX FTP Client for a Target Application	14
9. Running the NetX FTP Client Application Project	14
10. Conclusion.....	15
11. Next Steps.....	16
12. Reference Information	16

1. NetX and NetX Duo FTP Client Features

- Multi-thread support
 - High level APIs
 - Connecting and disconnecting
 - Directory operations
 - File operations

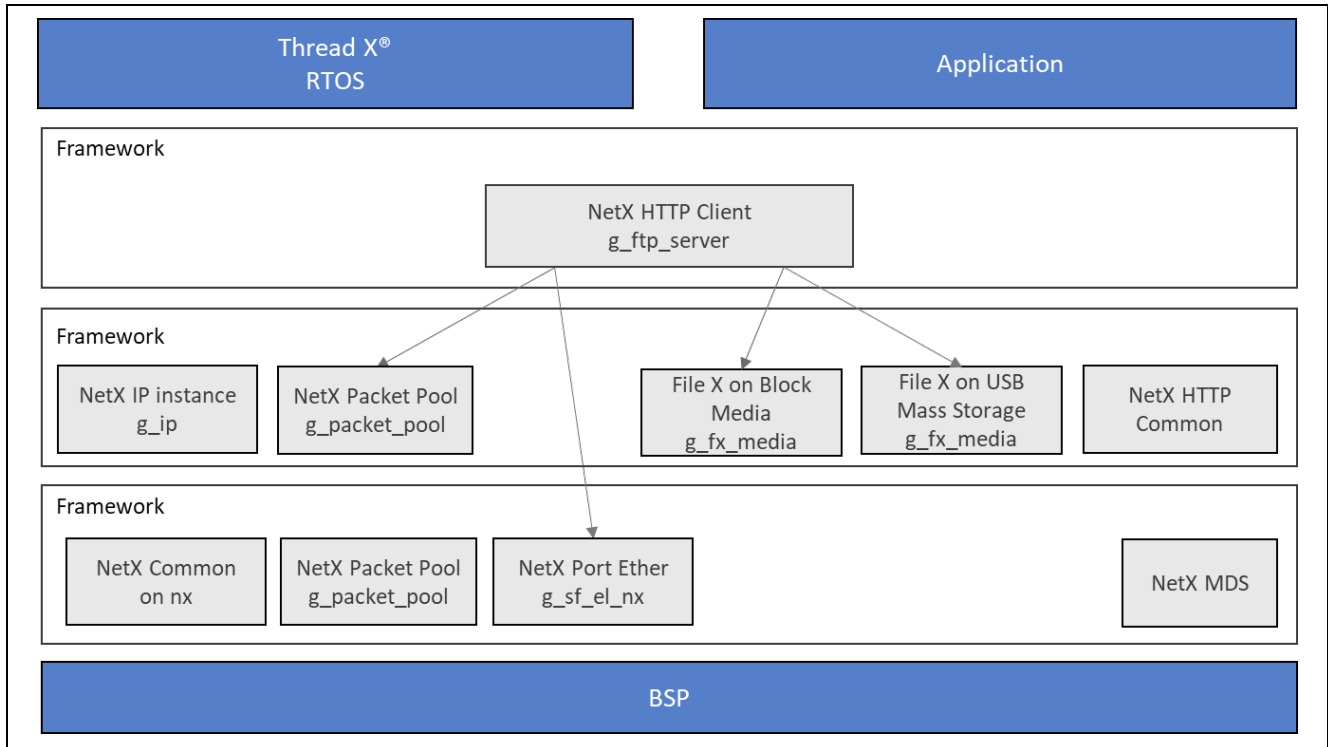


Figure 1 NetX FTP Client Organization, Options and Stack Implementations

2. NetX and NetX Duo FTP Client APIs Overview

The NetX FTP Client and NetX Duo FTP Client define API's for client creation/deletion, connecting/disconnecting to/from the server, directory operations and file operations. 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 return status values follow.

Table 1 NetX FTP Client API Summary

Function Name	Example API Call and Description
.nx_ftp_client_create	<code>nx_ftp_client_create(&my_client, "My Client", &client_ip, 2000, &client_pool);</code> Delete an FTP Client instance.
.nx_ftp_client_delete	<code>nx_ftp_client_delete(&my_client);</code> Delete an FTP Client instance.
.nx_ftp_client_connect	<code>nxd_ftp_client_connect(&my_client, server_ip_addr, NULL, NULL, 100);</code> Connect to an FTP Server with IPv6 and IPv4 support.
.nxd_ftp_client_connect	<code>nxd_ftp_client_connect(&my_client, server_ip_addr, NULL, NULL, 100);</code> Connect to an FTP Server with IPv6 and IPv4 support.
.nx_ftp_client_disconnect	<code>nx_ftp_client_disconnect(&my_client, 200);</code> Disconnect from the FTP Server.
.nx_ftp_client_create	<code>nx_ftp_client_directory_create(&my_client, "my_dir", 200);</code> Create a directory on the server.
.nx_ftp_client_directory_set	<code>nx_ftp_client_directory_listing_set(&my_client, "my_dir", &my_packet, 200);</code> Get a directory listing from the server.
.nx_ftp_client_directory_delete	<code>nx_ftp_client_directory_delete(&my_client, "my_dir", 200);</code> Delete a directory on the server.
.nx_ftp_client_directory_listing_get	<code>nx_ftp_client_directory_listing_get(&my_client, "my_dir", &my_packet, 200);</code> Get a directory listing from the server.
.nx_ftp_client_directory_listing_continue	<code>nx_ftp_client_directory_listing_continue(&my_client, &my_packet, 200);</code> Continue a directory listing from the server.
.nx_ftp_client_file_open	<code>nx_ftp_client_file_open(&my_client, "my_file.txt", NX_FTP_OPEN_FOR_READ, 200);</code> Open a file on the server.
.nx_ftp_client_file_close	<code>nx_ftp_client_file_close(&my_client, 200);</code> Close the currently open file on the server.
.nx_ftp_client_file_read	<code>nx_ftp_client_file_read(&my_client, &my_packet, 200);</code> Read from a file already open on the server.
.nx_ftp_client_file_write	<code>nx_ftp_client_file_write(&my_client, my_packet, 200);</code> Write to an already open file on the server.
.nx_ftp_client_file_rename	<code>nx_ftp_client_file_rename(&my_client, "my_file.txt", "new_file.txt", 200);</code> Rename a file on the server.
.nx_ftp_client_file_delete	<code>nx_ftp_client_file_delete(&my_client, "my_file.txt", 200);</code> Delete a file on the server.

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 Reference Manual listed in the Reference Section at the end of this document.

Table 2 Error Status Return Values

Name	Description
NX_SUCCESS	Successful FTP function.
NX_FTP_NOT_DISCONNECTED	FTP client is already connected.
NX_FTP_200_CODE_NOT_RECEIVED	Server rejects FTP connection.
NX_FTP_300_CODE_NOT_RECEIVED	Server rejects user/password.
NX_PTR_ERROR	Invalid FTP, username, or password pointer.
NX_CALLER_ERROR	Invalid caller of this service.
NX_IP_ADDRESS_ERROR	Invalid IP address.
NX_FTP_NO_2XX_RESPONSE_XXD	FTP server error response.
NX_FTP_NO_2XX_RESPONSE_PORT	FTP server response to PORT.
NX_FTP_NO_1XX_RESPONSE	FTP server response to NLST.
NX_FTP_END_OF_LISTING	No more entries in directory.
NX_FTP_NO_2XX_RESPONSE_DISCONNECT	FTP server response to disconnect.

Note: Lower level drivers may return Common Error Codes. Refer to the *SSP User's Manual* listed in the Reference Section at the end of this document for a definition of all relevant Error codes.

3. NetX and NetX Duo FTP Client Operational Overview

FTP Client Requirements

To function properly, the NetX FTP Client package requires NetX. Similarly, the NetX Duo FTP Client package relies on NetX Duo. The host application must create an IP instance for running NetX services and periodic tasks. If running the FTP host application over an IPv6 network, IPv6, and ICMPv6 must be enabled on the IP task. TCP must be also enabled for either IPv6 or IPv4 networks. The IPv6 host application must set its link local and global IPv6 address using the IPv6 API and/or DHCPv6.

The FTP Server and Client are 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.

The FTP Client portion of the NetX FTP package has no further requirements.

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 FTP implementation. The packet pool payload size should be able to accommodate the maximum path and file name.

FTP Client Commands

The FTP has a simple mechanism for opening connections and performing file and directory operations. There is basically a set of standard FTP commands that are issued by the Client after a connection has been successfully established on the TCP well-known port 21. The following shows some of the basic FTP commands. Note that the only difference when FTP runs over IPv6 is that the PORT command is replaced with the EPRT command:

Table 3 FTP Client 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

Note: These ASCII commands are used internally by the NetX FTP Client software to perform FTP operations with the FTP Server.

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
0xx	Syntax error in command.
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.

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 Client must provide the Server with a username and password. Some FTP sites allow what is called Anonymous FTP, that 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.

You are responsible for supplying the NetX FTP Client and the NetX Duo FTP Client 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 the FTP operations are allowed. Otherwise, if the login function returns something other than `NX_SUCCESS`, the connection attempt is rejected.

3.1 Important Operational Notes and Limitations

3.1.1 Operational Notes

FTP Multi-Thread Support

- The NetX FTP Client services can be called from multiple threads simultaneously. However, read or write requests for a particular FTP Client instance should be done in sequence from the same thread.

RFCs

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

3.1.2 Limitations

FTP Constraints

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

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

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

4. Including the NetX FTP and NetX Duo FTP Client in an Application

This section describes how to include the NetX FTP Client 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 *Getting Started Guide for SSP* listed in the Reference Section at the end of this document to learn how to manage each of these important steps in creating SSP based applications.

To add the NetX FTP Client to an application, simply add it to a thread using the Stacks Selection Sequence given in the table below. (The default name for the NetX FTP Client is `g_ftp_client0` and this is shown in the below table. This name can be changed in the associated Properties window). When the driver is added it shows up in the thread as illustrated in Figure 2, below.

Use the below table as a starting point for format, font and sequence.

Table 6 NetX FTP Client Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
<code>g_ftp_client0</code> NetX FTP Client	Threads	New Stack> X-Ware> NetX> Protocols> NetX FTP Client

When the NetX FTP Client on NetX FTP Client is added to the Thread Stack, as shown in the figure below, 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.

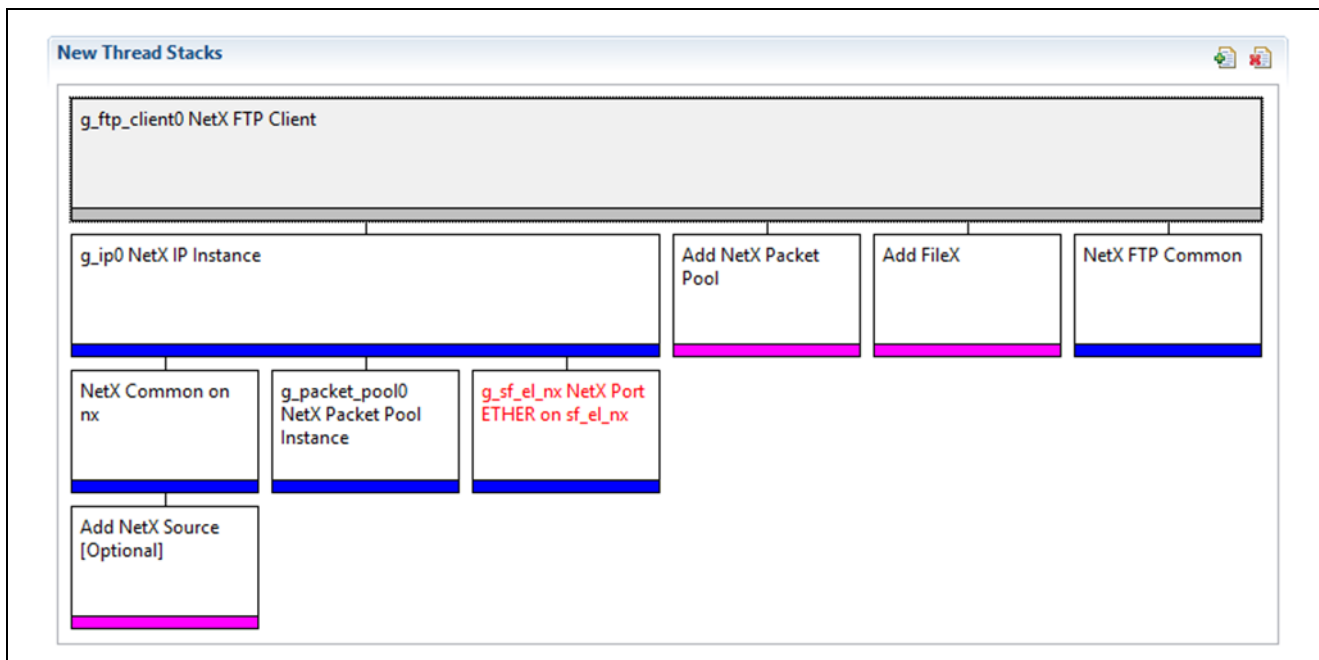


Figure 2 NetX FTP Client Stack

5. Configuring the NetX and NetX Duo FTP Client Module

The NetX FTP Client and the NetX Duo FTP Client modules must be configured by the user for the desired operation. The SSP configuration window will automatically identify, by highlighting the block in red, any required configuration

selections, such as Interrupts or operating modes, which must be configured for lower level modules, for successful operation. Furthermore, 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 Property 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 with 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. Also 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 below configuration properties tables, but is easily visible with the ISDE when configuring Interrupt Priority levels.

Note: You may want to open your ISDE and create the NetX FTP Client or the NetX Duo FTP Client and explore the property settings in parallel with looking over the Configuration Table Settings given below. 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 NetX FTP Client

ISDE Property	Setting	Description
Name	g_ftp_client0	Module name.
TCP socket window size (bytes)	2048	TCP socket window size selection.

Note: The above setting examples and defaults are for a project using the Synergy MCU Family. 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 pins for the Ethernet peripheral. The configurable properties for the lower level stack modules are given in the below sections for completeness and as a reference.

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

5.1 Configuration Settings for the NetX and NetX Duo FTP Client Low Level Drivers

Only a small number of settings must be modified from the default for lower level drivers and these are indicated via 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 table below identifies all the settings within the properties section for the module.

Table 8 Configuration for the NetX IP Instance

ISDE Property	Setting	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)	2048	IP Helper Thread Stack Size (bytes) selection
IP Helper Thread Priority	3	IP Helper Thread Priority selection
ARP	Enable	ARP selection
ARP Cache Size in Bytes	512	ARP Cache Size in Bytes selection
Reverse ARP	Enable, Disable (Default: Enable)	Reverse ARP selection
TCP	Enable	TCP selection
UDP	Enable, Disable (Default: Enable)	UDP selection
ICMP	Enable, Disable (Default: Enable)	ICMP selection
IGMP	Enable, Disable (Default: Enable)	IGMP selection

Table 9 Configuration for the NetX Packet Pool Instance on g_packet_pool1

ISDE Property	Setting	Description
Name	g_packet_pool1	Module name
Packet Size in Bytes	512	Packet size selection
Number of Packets in Pool	16	Number of packets in pool selection

Table 10 Configuration for the FileX on Block Media

ISDE Property	Setting	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.
Formatting Options		Formatting options selection.
Volume Name	Volume 1	Volume name selection.
Number of FATs	1	Number of FATs 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 11 Configuration for the FileX on USB Mass Storage

ISDE Property	Setting	Description
Name	g_fx_media0	Module name.

Table 12 Configuration for NetX 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 13 Configuration for NetX Common on nx

ISDE Property	Setting	Description
No configurable properties		

Table 14 Configuration for NetX Packet Pool Instance on g_packet_pool0

ISDE Property	Setting	Description
Name	g_packet_pool0	Module name
Packet Size in Bytes	512	Packet size selection
Number of Packets in Pool	16	Number of packets in pool selection

Table 15 Configuration for NetX 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_07_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	Priority 0(highest)-15(lowest), Disabled (Default: Priority 5)	Ethernet interrupt priority selection
Name	g_sf_el_nx	Module name
Channel	0	Channel selection
Callback	NULL	Callback selection

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

5.2 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 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. Table 7 illustrates the method for selecting the pins within the SSP configuration window and Table 8 illustrates an example selection for the I2C pins.

Note that the Operation Mode selection mode determines what peripheral signals are available and thus what MCU pins are required.

Table 16 Pin Selection Sequence for ETHERC Module

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

Note: The above selection sequence assumes ETHERC1 is the desired hardware target for the driver.

Table 17 Pin Configuration Settings for ETHERC1

Pin Configuration Property	Settings	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

Note: The above example settings are for a project using the S7G2 Synergy MCU and the SK-S7G2 Kit. Other Synergy Kits and other Synergy MCUs may have different available pin configuration settings.

6. Using the NetX FTP Client in an Application

The below example assumes a local FTP Server is already running. The typical steps in using the NetX FTP Client in an application are:

1. Create FTP Client using nx_ftp_client_create
2. Connect to the FTP Server using the nx_ftp_client_connect API
3. Open a file using the nx_ftp_client_open API

Write to a file using the nx_ftp_client_write API

1. Close a file using the nx_ftp_client_close API
2. Open a file using the nx_ftp_client_open API
3. Read from a file using the nx_ftp_client_read API
4. Close a file using the nx_ftp_client_close API

The above common steps are illustrated in a typical operational flow diagram in Figure 3 below.

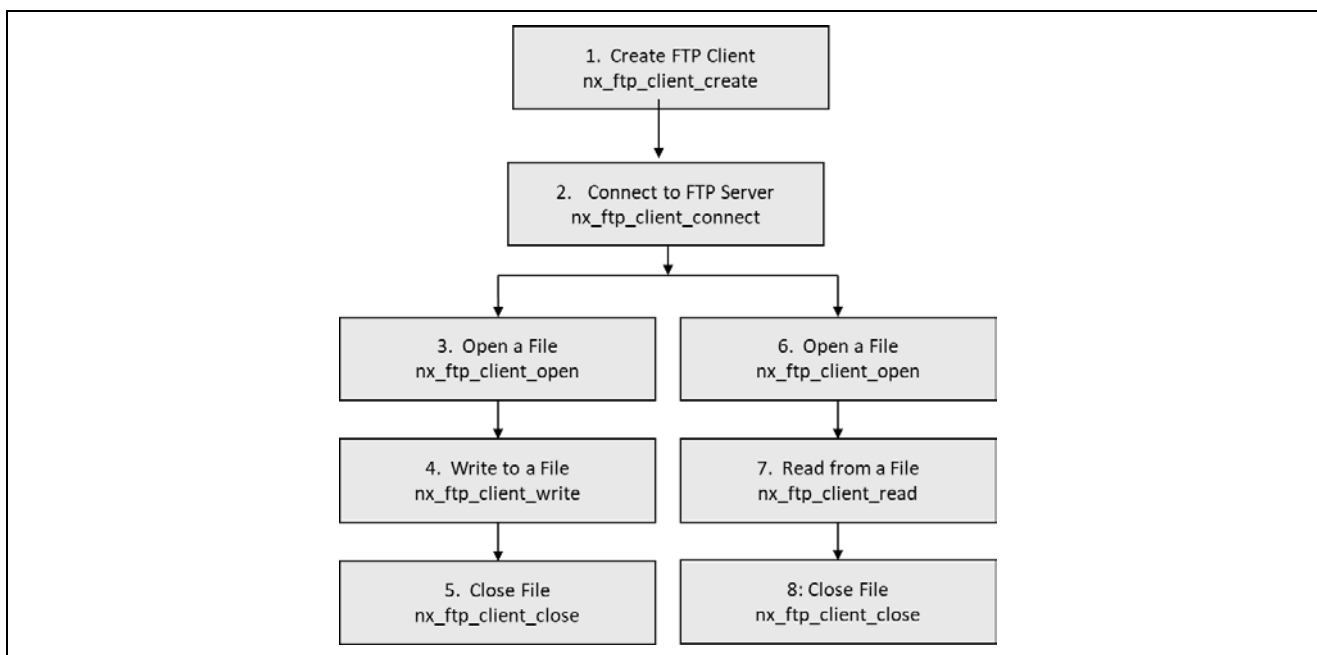


Figure 3 Flow Diagram of a Typical NetX FTP Client Application

7. The NetX FTP Client Application Project

The Application Project associated with this Module Guide demonstrates the above steps in a full design. The project can be found using the link provided in the Reference Section at the end of this document. You may want to import and open the Application Project within ISDE and view the configuration settings for the NetX FTP Client Module. You can also read over the code, in `ftp_client_thread_entry.c` and `console_thread_entry.c`, which is used to illustrate the NetX FTP Client API's in a complete design.

The Application Project demonstrates the typical use of the NetX FTP Client APIs. The Application Project FTP Client thread entry initializes the DHCP client and waits until an IP address is acquired. It then waits until USB storage device is attached to the USB port. The loop in the thread blinks LED3 to indicate activity. Actual usage of the FTP Client is done from the console; this is implemented in the Console thread.

The Console thread uses the Console Framework to provide a simple interactive example of use of the FTP Client. For more information on the Console Framework read the corresponding Module Guide and SSP documentation. The Console thread prompts for commands and reacts to them. There are commands for specifying the FTP server, user name and password; to connect and disconnect; and to list files on the FTP Server.

The table below identifies the target versions for the associated software and hardware used by the Application Project.

Table 18 Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e ² studio	5.3.1.002	Integrated Solution Development Environment
SSP	1.2.0	Synergy Software Platform
DK-S7G2	REV2	Development Kit

The complete Application Project can be found using the link provided in the Reference Section at the end of this document. The `ftp_client_thread_entry.c` file is located in the project once it has been imported into the ISDE. You can open this file, within the ISDE, and follow along with the below description, to help identify key uses of APIs.

8. Customizing the NetX FTP Client for a Target Application

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

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

The FTP Client 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.

9. Running the NetX FTP Client Application Project

To run the NetX FTP Client Application project and to see it executing on a target kit, you can simply import it into your ISDE, compile and run debug.

To implement the NetX FTP Client application in a new project, follow the steps below.

Note: The below 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 “Getting Started with SSP” Chapter in the “SSP User’s Manual”, listed in the Reference Section at the end of this document.

To create and run the NetX FTP Client Application Project simply follow these steps:

1. Refer to the Synergy Project Import Guide (11an0023eu0114-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 with an active DHCP server.
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. Connect the USB_FS port (micro USB) to a computer that will be used as console.
6. Start to debug the application.
7. Debug output can be viewed in the Renesas Debug Console.
8. When the application is first started the necessary drivers for the console may need to be installed. Follow the steps necessary to accomplish this in your system.
9. Once the communication port is ready open a terminal program and connect to the console. In the example screen below, Tera Term is used.
10. Once connected to the console type “?” (no quotes) and enter to display the help.
11. Set the server IP address (dotted decimal notation, for example 192.168.2.103), the username and password for the FTP server used for testing.
12. Use the CONNECT command to connect to the server.
13. Use the LIST [<directory>] command to list the default or a specific directory on the server.
14. A screen shot of an example session is shown below.

```
COM7 - Tera Term VT
File Edit Setup Control Window Help
Command>
Command?
Command Help Menu
SERVER : Enter the FTP address
USER : Enter the FTP user
PASS : Enter the FTP password
CONNECT : Connect to the FTP server
DISCONNECT : Disconnect from the FTP server
LIST : List files on the FTP server

Command>SERVER 192.168.2.103
SERVER: "0xC0A80267"
Command>USER test
USER: "test"
Command>PASS 123
PASS: "123"
Command>CONNECT
Connecting...
SERVER: "0xC0A80267"
USER: "test"
PASS: "123"
Connected!
Command>LIST
Packet length: 12
Command>LIST /
Packet length: 13
/README.txt

Command>DISCONNECT
Disconnecting...
Disconnected!
Command>
```

Figure 4 Example of a session using the FTP client over the console

10. 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 incorrect selection of low level drivers. The use of high level API's, as demonstrated in the Application Project illustrate additional development time savings by allowing work to begin at a high level, avoiding the time required in older development environments to use or, in some cases, create low level drivers.

An FTP Client allows you to inspect and retrieve files from FTP Servers. The File Transfer Protocol is standardized and widely supported. It is a convenient light way of gaining network access to files. As seen in this note, the use of the FTP Client module requires only a few steps and provides full capabilities.

11. Next Steps

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

The demo does not get or put files. This can be accomplished by copying the LIST command and doing the small necessary modifications to it.

Consequently, also the USB mass storage is not used in this demo, but it is properly configured and ready to be used once the file put/get operations are used.

Other Application Projects and Application Notes that demonstrate FTP Client use can be found as described at the end of the Reference section at the end of this document.

12. Reference Information

1. [Start Developing with the Renesas Synergy™ Platform](#)
Provides an overview of each of the elements of the Renesas Synergy Platform
2. [SSP User's Manual](#)
A detailed reference manual for all SSP modules, APIs, structures and error codes.
3. *NetX™ and NetX Duo™ FTP Client Module Guide*
The application project associated with this Module Guide
4. [How do I Use Printf\(\) with the Debug Console in the Synergy Software Package](#)
Provides step by step process to add printf() with the debug console in the SSP.

Website and Support

Support: <https://synergygallery.renesas.com/support>

Technical Contact Details:

- America: <https://www.renesas.com/en-us/support/contact.html>
- Europe: <https://www.renesas.com/en-eu/support/contact.html>
- Japan: <https://www.renesas.com/ja-jp/support/contact.html>

All trademarks and registered trademarks are the property of their respective owners.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jan 10, 2018	-	Initial version

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.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

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

(Rev.4.0-1 November 2017)



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.

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

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: +44-1628-651-700, Fax: +44-1628-651-804

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, 200333 P. R. China
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 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.

17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338