

Micrium

Empowering Embedded Systems

μ C/TCP-IP V1.85

Release Notes

www.Micrium.com

Revision History

Version	Date	Description
V1.85	2006 Jun	Bug fixes & enhancements
V1.84	2006 Apr	Bug fixes & enhancements
V1.83	2005 Dec	Bug fixes & enhancements
V1.82	2005 Dec	Bug fixes & enhancements
V1.81	2005 Oct	New features & enhancements
V1.80	2005 Oct	Network communication performance improvements Bug fixes & enhancements
V1.73	2005 Aug	TCP transmit round-trip time & retransmission controls Bug fixes & enhancements
V1.72	2005 Jul	Bug fixes & enhancements
V1.71	2005 May	Bug fixes & enhancements
V1.70	2005 Apr	TCP transmit congestion and window controls Bug fixes & enhancements
V1.61	2005 Feb	Bug fixes
V1.60	2005 Feb	TCP receive congestion and window controls First version with release history
V1.56	2004 Dec	Bug fixes & enhancements
V1.54	2004 Dec	Bug fixes & enhancements
V1.52	2004 Nov	Bug fixes & enhancements
V1.50	2004 Oct	First TCP/IP version released
V1.00	2004 Jun	First Beta version released
V0.60	2004 Feb	First UDP/IP version released
V0.50		

Requires the following versions of needed Modules

Version 1.85

μC/CPU Version 1.14

μC/LIB Version 1.20

Version 1.84

μC/CPU Version 1.13

μC/LIB Version 1.18

Version 1.83

μC/CPU Version 1.13

μC/LIB Version 1.18

Version 1.82

μC/CPU Version 1.13

μC/LIB Version 1.18

Version 1.81

μC/CPU Version 1.13

μC/LIB Version 1.18

Version 1.80

μC/CPU Version 1.13

μC/LIB Version 1.18

Version 1.73

μ C/CPU Version 1.12

μ C/LIB Version 1.17

Version 1.72

μ C/CPU Version 1.12

μ C/LIB Version 1.17

Version 1.71

μ C/CPU Version 1.11

μ C/LIB Version 1.15

Version 1.70

μ C/CPU Version 1.11

μ C/LIB Version 1.14

Version 1.61

μ C/CPU Version 1.10

μ C/LIB Version 1.13

Version 1.60

μ C/CPU Version 1.10

μ C/LIB Version 1.13

New Features

Version 1.85

N/A

Version 1.84

V1.84-001

Added new IP status functions :

<code>NetIP_GetAddrThisHost()</code>	returns the currently configured IP address for this host.
<code>NetIP_GetAddrDfltGateway()</code>	returns the currently configured IP address of this host's default gateway.
<code>NetIP_GetAddrStrThisHost()</code>	returns the currently configured IP address for this host, as an ASCII string.
<code>NetIP_GetAddrStrDfltGateway()</code>	returns the currently configured IP address of this host's default gateway, as an ASCII string.

Version 1.83

N/A

Version 1.82

N/A

Version 1.81

V1.81-001

Added new TCP configuration function :

<code>NetTCP_ConnCfgTxAckImmedRxdPushEn()</code>	configures TCP connection to enable/disable transmitting immediate acknowledgements for any received & pushed TCP segments.
--	---

Version 1.80

V1.80-001

Added new Network Debug Status Functions :

NetDbg_ChkStatusTCP ()	returns TCP layer status.
NetDbg_ChkStatusBufs ()	returns network buffer(s) status.

Version 1.73

V1.73-001

TCP Transmit Congestion Controls

- Round-Trip Time (RTT)
- Retransmission Timeout (RTO)

Version 1.72

N/A

Version 1.71

V1.71-001

Added Network Debug Status Functions :

NetDbg_ChkStatus ()	returns network status errors & faults.
NetDbg_ChkStatusRsrcLost ()	returns network resource(s) lost status.
NetDbg_ChkStatusRsrcLo ()	returns network resource(s) low status.
NetDbg_ChkStatusConns ()	returns network connection(s) status.

Version 1.70

V1.70-001

TCP Transmit Congestion Controls

- Slow Start / Congestion Avoidance
- Fast Re-transmit / Fast Recovery
- Nagle Algorithm
- Transmit Silly Window Syndrome Avoidance
- Transmit Window Configuration / Control
- Idle Connection / Zero-Window Probe Persist Timers

Version 1.61

N/A

Version 1.60

V1.60-001

TCP Receive Congestion Controls

- Immediate & Delayed Acknowledgements
- Receive Silly Window Syndrome Avoidance
- Receive Window Configuration / Control

V1.60-002

TCP Transmit Segment Aggregation

- Data
- FIN-Close
- PUSH

Improvements

Version 1.85

V1.85-001

Improved ARM assembly port files to be compatible for both ARM & Thumb modes.

Version 1.84

V1.84-001

Improved TCP/Socket/Connection Close Management to reduce/eliminate connection leaks, corruption, faults.

V1.84-002

NetTCP_ConnClose() modified to transmit immediate TCP reset packet on any abnormal or fault TCP connection close.

Version 1.83

V1.83-001

Added BSD configuration, NET_BSD_CFG_API_EN to indicate whether BSD 4.x Layer API functionality should be enabled & included in the project build :

DEF_DISABLED	BSD 4.x API functions disabled & excluded from build
DEF_ENABLED	BSD 4.x API functions enabled & included in build

Version 1.82

V1.82-001

Refactored network buffer free functionality into appropriate NetBuf_FreeBuf() functions :

NetBuf_FreeBuf()	free a network buffer.
NetBuf_FreeBufList()	free a network buffer list.
NetBuf_FreeBufQ_PrimList()	free a network buffer queue, organized by primary list.
NetBuf_FreeBufQ_SecList()	free a network buffer queue, organized by secondary list.

V1.82-002

Improved NetDbg_ChkStatus() functions by returning more specific debug status codes.

Version 1.81

V1.81-001

Improved NetTCP_ConnCfg() functions' validation & configuration.

Version 1.80

V1.80-001

Network Transmit/Receive Load Balancing

- Handle network transmit & receive packets in an approximately balanced ratio.

V1.80-002

Improved NetIP_TxPktDatagramNextHopSel()'s IP address host, network, & default gateway validation.

Version 1.73

V1.73-001

Added NIC configuration, NET_NIC_CFG_TX_PKT_PREPARE_EN to allow the NIC to prepare transmit packet(s) while simultaneously transmitting previously-prepared packets.

If configured as DEF_ENABLED, the NIC driver **MUST** implement an appropriate NetNIC_TxPktPrepare() function to prepare transmit packet(s) separate from transmitting them in NetNIC_TxPkt().

Version 1.72

V1.72-001

Added NIC configuration, NET_NIC_CFG_RD_WR_SEL to indicate how the NIC's read/write functionality should be implemented :

NET_NIC_RD_WR_SEL_MACRO	implement with macro's to improve read/write performance
NET_NIC_RD_WR_SEL_FNCT	implement with functions to handle more complex read/write functionality

Version 1.71

V1.71-001

Added fatal Socket error code, `NET_SOCKET_ERR_FAULT`. When this or the following error codes are returned by a socket function, that socket must be immediately `close()`'d with no further access :

```
NET_SOCKET_ERR_NOT_USED
NET_SOCKET_ERR_INVALID_FAMILY
NET_SOCKET_ERR_INVALID_PROTOCOL
NET_SOCKET_ERR_INVALID_TYPE
NET_SOCKET_ERR_INVALID_STATE
NET_SOCKET_ERR_FAULT
```

Version 1.70

N/A

Version 1.61

N/A

Version 1.60

V1.60-001

Modified String Conversion Functions Argument Lists :

```
ASCII MAC address to MAC address: NetASCII_Str_to_MAC( )
MAC address to ASCII MAC address: NetASCII_MAC_to_Str( )
```

NOTE: New arguments were added to these functions.

V1.60-002

Improved IP address String Conversion Functions :

```
NetASCII_Str_to_IP( ) now allows leading zeros in IP address string.
NetASCII_IP_to_Str( ) now allows ASCII IP address to have leading zeros.
```

Changes

Version 1.85

V1.85-001

Added 'NET_UDP_MTU_ACTUAL' & 'NET_TCP_MTU_ACTUAL' to adjust the transport layer Maximum Transmission Units (MTUs) based on the configured values for network buffer sizes.

Version 1.84

V1.84-001

Changed Debug information & status names to remove redundant 'DBG' token. E.g. 'NET_DBG_CFG_DBG_INFO_EN' to 'NET_DBG_CFG_INFO_EN' & 'NET_DBG_CFG_DBG_STATUS_EN' to 'NET_DBG_CFG_STATUS_EN'.

Version 1.83

N/A

Version 1.82

N/A

Version 1.81

V1.81-001

NetTCP_TxConnAck() modified to transmit immediate acknowledgments for any received & pushed TCP segments based on the configuration of a TCP connection enable (see also 'New Features V1.81-001').

Version 1.80

V1.80-001-a

Changed all `net_cfg.h` timer configurations from previously configuring timers with values & units of network ticks to configuring values & units of seconds or milliseconds :

```
NET_TCP_CFG_TIMEOUT_CONN_MAX_SEG_SEC
NET_TCP_CFG_TIMEOUT_CONN_RX_Q_SEC
NET_TCP_CFG_TIMEOUT_CONN_TX_Q_SEC
NET_SOCKET_CFG_TIMEOUT_RX_Q_SEC
NET_SOCKET_CFG_TIMEOUT_CONN_REQ_SEC
NET_SOCKET_CFG_TIMEOUT_CONN_ACCEPT_SEC
NET_SOCKET_CFG_TIMEOUT_CONN_CLOSE_SEC
```

V1.80-001-b

Changed `net_os.c` timer configuration functions from configuring timers in units of network ticks to configuring timers in units of seconds or milliseconds.

Examples :

```
void NetOS_TxSuspendTimeoutSet(CPU_INT32U    timeout_ms,
                               NET_ERR        *perr);

void NetOS_Sock_RxQ_TimeoutSet(NET_SOCKET_ID sock_id,
                               CPU_INT32U    timeout_sec,
                               NET_ERR        *perr);
```

V1.80-001-c

Changed all `net_os.c` timer configuration functions from returning timeout values in units of network ticks to returning timeout values in units of milliseconds :

```
CPU_INT32U NetOS_TxSuspendTimeoutGet_ms(NET_ERR *perr);

CPU_INT32U NetOS_Sock_RxQ_TimeoutGet_ms(NET_SOCKET_ID sock_id,
                                         NET_ERR        *perr);
```

V1.80-002

Added `NET_CFG_TX_SUSPEND_TIMEOUT_MS` to `net_cfg.h` to configure the network transmit/receive load balancing timeout.

Version 1.73

N/A

Version 1.72

N/A

Version 1.71

V1.71-001-a

Moved Network Status Monitor Task from ICMP Module to Network Debug Module.

V1.71-001-b

Moved ICMP Monitor Low Resource configuration to Network Debug Module:

```
NetDbg_CfgRsrcBufSmallThLo( )  
NetDbg_CfgRsrcBufLargeThLo( )  
NetDbg_CfgRsrcTmrThLo( )  
NetDbg_CfgRsrcConnThLo( )  
NetDbg_CfgRsrcARP_CacheThLo( )  
NetDbg_CfgRsrcTCP_ConnThLo( )  
NetDbg_CfgRsrcSockThLo( )
```

V1.71-001-c

Changed ICMP Monitor Task configuration to ICMP Transmit Source Quench configuration.
I.e. 'NET_ICMP_CFG_MON_TASK_EN' to 'NET_ICMP_CFG_TX_SRC_QUENCH_EN'.

Version 1.70

N/A

Version 1.61

N/A

Version 1.60

N/A

Corrections

Version 1.85

V1.85-001a

`NetConn_CloseFromApp()` & `NetConn_CloseFromTransport()` failed to check for previously-closed network connections following any other network connection close operations. Fixed by checking for & skipping any previously-closed network connections.

V1.85-001b

`NetConn_CloseAllConnListConns()` failed to properly check for asynchronously-freed network connections while closing all network connections from a network connection list. Fixed by checking for & advancing past any asynchronously-freed network connections in the network connection list.

V1.85-002

`NetDbg_ChkStatus()` status functions failed to acquire the global network lock for asynchronous access by applications. Fixed by acquiring the global network lock.

Version 1.84

V1.84-001a

`NetTCP_RxPktConnHandlerTxWinRemote()` incorrectly allowed received duplicate acknowledgement segments to update the remote receive window. Fixed by preventing duplicate acknowledgement segments from updating the remote receive window.

V1.84-001b

`NetTCP_TxConnWinSizeHandlerCongCtrl()` failed to compensate the remote receive window update by the TCP connection's recently transmitted sequences. Fixed by compensating the remote receive window update by the number of recently transmitted sequences.

V1.84-002

`NetTCP_TxConnTxAck()` incorrectly invalidated received synchronization (SYN) or close (FIN) segments with no data as acknowledgement-only segments & therefore failed to transmit a TCP acknowledgement segment in reply. Fixed by validating received synchronization or close segments as valid TCP control segments that require acknowledgement.

Version 1.83

V1.83-001a

NetTCP_RxPktConnHandlerRxQ_Conn() failed to properly trim duplicate sequences prior to the next expected receive sequence for received segments whenever the TCP connection's transport receive queue was initially empty. Fixed by handling all received segments similarly, thereby properly trimming all duplicate sequences regardless of whether the transport receive queue is initially empty or non-empty.

V1.83-001b

NetTCP_RxPktConnHandlerRxQ_Conn() incorrectly cast received segments' buffer sequence numbers to 16-bit instead of 32-bit. Fixed by casting to 32-bit.

V1.83-001c

NetTCP_RxPktConnHandlerRxQ_Conn() incorrectly decremented received duplicate data segments' 'TCP_SegLen' only without also decrementing 'TCP_SegLenData'. Fixed by also decrementing 'TCP_SegLenData'.

Version 1.82

V1.82-001

NetTCP_TxConnReTxQ() failed to properly prepare & re-transmit certain partially acknowledged segments in the re-transmit queue whose total transmit packet length was smaller than the minimum network interface packet size. Fixed by properly preparing all TCP re-transmit segments for the minimum network interface packet size.

V1.82-002

NetTmr_TaskHandler() failed to properly check for asynchronously-freed timers while handling timers on the Timer Task List. Fixed by checking for & advancing past any asynchronously-freed timers in the Timer Task List.

Version 1.81

N/A

Version 1.80

V1.80-001

`NetTCP_RxAppData()` incorrectly closed TCP connections that received TCP close segments with no TCP data. Fixed by correctly handling all received TCP close segments, with and without TCP data.

V1.80-002

`NetTCP_RxPktConnIsValidSeq()` incorrectly invalidated duplicate received TCP close segments. Fixed by appropriately handling received TCP close segments prior to validating the TCP close segment sequence numbers.

V1.80-003

`NetTCP_RxPktConnHandlerSeg()` did not appropriately call `NetTCP_RxPktConnHandlerReTxQ()` for received duplicate acknowledgements which also contained a TCP close flag. This prevented TCP connections in certain states from closing. Fixed by correctly handling all received TCP close segments, regardless of duplicate acknowledgements.

V1.80-004

`NetTCP_RxPktConnHandlerRxQ_AppData()` incorrectly discarded certain received TCP synchronization segments with TCP data. Fixed by appropriately handling all received TCP synchronization segments, with or without TCP data.

V1.80-005

`NetTCP_RxPktConnHandlerReTxQ()` incorrectly validated partially acknowledged segments in the re-transmit queue. Partially acknowledged segments are acknowledged to an aligned number of sequences to avoid data alignment errors (see 'Corrections V1.72-002'), but a partial acknowledgement delta must be used to validate new acknowledgement segments. Fixed by correctly validating partially acknowledged segments using the partial acknowledgement delta.

V1.80-006

`NetTCP_TxPktHandler()` incorrectly returned transitory transmit errors for TCP segments that were discarded. Fixed by returning fatal transmit errors.

V1.80-007

`NetTCP_TxConnTxQ()` incorrectly used only the amount of queued TCP transmit data octets in validating the next transmit segment for the Nagle algorithm. The algorithm should use the minimum of the queued TCP transmit data amount and the actual length of the next TCP segment to transmit. Fixed by correctly comparing and using the minimum of the queued TCP transmit data and the next TCP transmit segment length to validate the Nagle algorithm.

Version 1.73

V1.73-001

`inet_addr()` incorrectly returned `in_addr` structure. Fixed by returning `in_addr_t` IP address.

V1.73-002

`NetTCP_RxPktConnHandlerReTxQ()` failed to check the TCP re-transmit queue's updated head buffer as non-null before handling. Fixed by checking the re-transmit queue's new head buffer for non-null.

V1.73-003

Subnetted broadcast or specified host IP addresses incorrectly validated or handled. Fixed by correctly validating or handling these subnetted IP addresses.

Version 1.72

V1.72-001

ARP cache handler incorrectly incremented transmit buffers' reference counter when queuing transmit buffers to a pending ARP cache. Fixed by not incrementing transmit buffers' reference counters.

V1.72-002

`NetTCP_RxPktConnHandlerReTxQ()` incorrectly updated partially acknowledged segments in the re-transmit queue. Advancing the segments by a non-aligned number of sequences created data alignment errors. Fixed by correctly advancing partially acknowledged segments by an aligned number of sequences.

Version 1.71

V1.71-001

NetTCP_RxPktConnHandlerReTxQ() incorrectly updated the TCP re-transmit queue's initial head buffer instead of the new head buffer when updating partial segments. Fixed by updating the re-transmit queue's new head buffer.

V1.71-002

NetTCP_RxPktConnHandlerRxQ_Conn() used incorrect TCP connection receive window variable 'RxWinSizeActual' instead of 'RxWinSizeCfgdActual' to sequence received segments. Fixed by using the correct TCP receive window variable.

V1.71-003

NetTCP_RxConnWinSizeHandler() always transmitted an immediate TCP acknowledgement whenever a TCP connection's local receive window updated, even if the acknowledgement should have been delayed. Fixed by delaying the acknowledgement, when applicable.

V1.71-004

NetTCP_TxConnTxQ() did not clear a pointer when moving segments from TCP transmit queue to TCP re-transmit queue. Fixed by clearing the pointer.

V1.71-005

NetTCP_TxConnTxQ() always requested a transmit silly window timer, even if the TCP connection had already or previously requested a timer. Fixed by not getting a new transmit silly window timer if the TCP connection previously requested a timer.

V1.71-006-a

NetIP_RxPktValidate() did not invalidate received IP datagrams with 'This Host' or specified host destination addresses. Fixed by invalidating these destination addresses.

V1.71-006-b

NetIP_TxPktValidate() did not validate transmit IP datagrams with a broadcast destination address. Fixed by validating the broadcast address as a destination address.

V1.71-007

NetBuf_GetMaxSize() always returned the maximum buffer data size for large buffers, even if the current buffer was a small buffer. Fixed by returning the maximum buffer data size for the current buffer, when applicable.

Version 1.70

V1.70-001

Duplicate acknowledgement segments incorrectly updated TCP connections' last unacknowledged sequence number. Fixed by updating TCP connections' last unacknowledged sequence number with segments that only acknowledge new data.

Version 1.61

V1.61-001

Data pointers advanced by packet lengths incorrectly cast the packet length increment to 8-bit, regardless of packet length size. Fixed by removing incorrect 8-bit cast.

V1.61-002

Random port numbers NOT freed back to random port number queue for stream socket `close()`. Fixed by freeing random port numbers for all socket `close()`.

Version 1.60

V1.60-001

TCP timeouts not set on transition to TCP `close()`. Fixed by adding/updating TCP timeouts on transition to TCP `close()`.

Known Problems

Version 1.85

V1.73-001 (Unresolved)

Version 1.84

V1.73-001 (Unresolved)

V1.60-001 (Redesigned, Verification Ongoing;
see also 'Improvements V1.84-001')

Version 1.83

V1.73-001 (Unresolved)

V1.60-001 (Verification Ongoing)

Version 1.82

V1.73-001 (Unresolved)

V1.60-001 (Verification Ongoing)

Version 1.81

V1.73-001 (Unresolved)

V1.60-001 (Verification Ongoing)

Version 1.80

V1.73-001 (Unresolved)

V1.60-001 (Verification Ongoing)

Version 1.73

V1.73-001

Received IP broadcasts not demultiplexed to appropriate, non-wildcard-address socket(s).

V1.60-001 (Verification Ongoing)

Version 1.72

V1.60-001 (Verification Ongoing)

Version 1.71

V1.60-001 (Verification Ongoing)

Version 1.70

V1.60-001 (Verification Ongoing)

Version 1.61

V1.60-001 (Verification Ongoing)

Version 1.60

V1.60-001

TCP/Socket/Connection Close Updates:

Verify that all normal & fault connection closes, close correctly.

(See also 'Improvements V1.84-001')

Limitations

001

Only supports a single :

- (a) network interface (IF/NIC)
- (b) host IP address
- (c) default gateway

002

Following IP features NOT supported :

- (a) IP forwarding/routing
- (b) IP multicasting
- (c) IP transmit fragmentation
- (d) IP Security options
- (e) ICMP Address Mask Agent/Server

003

Following TCP features NOT supported :

- (a) TCP Urgent Data
- (b) TCP Security & Precedence
- (c) TCP Multihoming

004

Following socket features NOT supported :

- (a) Socket shutdown()
- (b) Multiple sockets bound to same socket address or socket pair

Contacts

Micrium

949 Crestview Circle

Weston, FL 33327

USA

+1 954 217 2036

+1 954 217 2037 (FAX)

e-mail: Licensing@Micrium.com

WEB: www.Micrium.com