API Function

Type of Functions

(1) Internal Function: Used inside the driver function
(2) API Function: Used in applications

Function Name	**void Int0(void) interrupt 0**
Arguments	None
Return value	None
Description	Interrupt handling function of the W3100A. Stores the status information that each function waits for in the global variable S_STATUS for transfer. S_STATUS stores the interrupt status value for each channel.
Category	Internal Function

Function Name	**void ISR_ESTABLISHED(SOCKET s)**
Arguments	s: Channel number
Return value	None
Description	Established connection interrupt handling function. Called upon connection establishment, and may be inserted in user code if needed by the programmer.
Category	Internal Function

Function Name	**void ISR_CLOSED(SOCKET s)**
Arguments	s: Channel number
Return value	None
Description	Closed connection interrupt handling function. Called upon connection closure, and may be inserted in user code if needed by the programmer.
Category	Internal Function

Function Name	**void ISR_RX(SOCKET s)**
Arguments	s: Channel number
Return value	None
Description	Received data interrupt handling function. Called upon receiving data, and may be inserted in user code if needed by the programmer.
Category	Internal Function

Function Name	**void initW3100A(void)**
Arguments	None
Return value	None
Description	W3100A initialization function. Function for S/W resetting of the W3100A. Sets the initial SEQ# to be used for TCP communication.
Category	API Function

Function Name	**void sysinit(u_char sbufsize, u_char rbufsize)**
Arguments	Sbufsize:tx memory size
Return value	rbufsize:rx memory size
Description	W3100A initialization function. Sets the source MAC, source IP, gateway, and subnet mask to be used by the W3100A to the designated values. May be called when setting the concerned register to modify network information and reflect it on the W3100A.
Category	API Function

Function Name	**void setsubmask(u_char * addr)**
Arguments	addr: Pointer having the value for setting up the subnet mask
Return value	None
Description	Subnet mask setup function
Category	API Function

Function Name	**void setgateway(u_char * addr)**
Arguments	addr: Pointer having the value for setting up the gateway IP
Return value	None
Description	Gateway IP setup function
Category	API Function

Function Name	**void setIP(u_char * addr)**
Arguments	addr: Pointer having the value for setting up the source IP address
Return value	None
Description	W3100A IP address setup function
Category	API Function

Function Name	**void setMACAddr(u_char * addr)**
Arguments	addr: Pointer having the value for setting up the MAC address
Return value	None
Description	MAC address setup function
Category	API Function

Function Name	**void settimeout(u_char * val)**
Arguments	val: Pointer having the value for setting up the timeout. Upper 2 bytes have the initial timeout value, while the last 1 byte has the number of retransmissions until timeout.
Return value	None
Description	TCP timeout setup function. Used for adjusting the TCP retransmission time. A timeout interrupt takes place when retransmission is attempted for establishing the connection or for data transfer beyond the set value.
Category	API Function

Function Name	**void setINTMask(u_char mask)**
Arguments	mask: Value of the mask to be set (‘1’ refers to interrupt enable)
Return value	None
Description	Interrupt mask setup function. Enables/disables the concerned interrupt.
Category	API Function

Function Name	**void setbroadcast(SOCKET s)**
Arguments	s: Channel number
Return value	None
Description	Broadcast data transfer enable setup function Enables/disables broadcasting data transfer in UDP or IP RAW mode.
Category	API Function

Function Name	**void setTOS(SOCKET s, u_char tos)**
Arguments	s: Channel number tos: Value to be set for the TOS field of the IP header
Return value	None
Description	Handles protocol setup function in IP RAW mode
Category	API Function

Function Name	**char socket(SOCKET s, u_char protocol, u_int port, u_char flag)**
Arguments	s: Channel number protocol: Protocol designated for the channel SOCK_STREAM(0x01) -> TCP SOCK_DGRAM(0x02) -> UDP SOCK_IPL_RAW(0x03) -> IP Layer RAW SOCK_MACL_RAW(0x04) -> MAC Layer RAW port: Source port designated for the channel flag: Options designated for the channel SOCKOPT_BROADCAST(0x80) -> '1' refers to broadcast data transfer in UDP mode SOCKOPT_NDTIMEOUT(0x40) -> '1' refers to use of only the register that designates the timeout value SOCKOPT_NDACK(0x20) -> '1' refers to the delayed ACK not to be used SOCKOPT_SWS(0x10) -> '1' refers to the silly window syndrome to be used
Return value	Channel number if succeeded, or –1 if failed.
Description	Initialization of the channel. Initializes the designated channel and waits for completion of W3100A handling.
Category	API Function

Function Name	**char connect(SOCKET s, u_char * addr, u_int port)**
Arguments	s: Channel number addr: Destination IP address port: Destination port number
Return value	1 if connection is established, or –1 if connection fails.
Description	Sets the connection to the designated peer. Establishes a connection with a peer on the designated channel and waits until the connection is established. (TCP client mode)
Category	API Function

Function Name	**char listen(SOCKET s, u_char * addr, u_int * port)**
Arguments	s: Channel number addr: Peer IP address at the time of connection establishment port: Peer Port number at the time of connection establishment
Return value	1 if connection is established, or –1 if connection fails.
Description	Waits for connection with a peer. (Blocking Mode) The designated channel waits for connection by a peer. (TCP Server mode)
Category	API Function

Function Name	char NBlisten(SOCKET s)
Arguments	s: Channel number
Return value	1
Description	Waits for connection with a peer. (Non-blocking Mode) The designated channel waits for connection by a peer. (TCP Server mode)
Category	API Function

Function Name	void initseqnum(SOCKET s)
Arguments	s: Channel number
Return value	None
Description	Generates random values for the initial SEQ# to be used for establishing a TCP connection. This function may be added to the code for generating random numbers for assigning a random number to initial SEQ# used in TCP. In an actual internet environment, the initial SEQ# must be a random number. (A fixed number is used for EVB/DK.)
Category	API Function

Function Name	u_int send(SOCKET s, u_char * buf, u_int len)
Arguments	s: Channel number buf: Pointer indicating the data to be sent len: Size of the data to be sent
Return value	Sent data size
Description	Function for sending TCP data. Composed of the send()and send_in() functions. The send() function is an application I/F function. It continues to call the send_in() function to complete the sending of the data up to the size of the data to be sent when the application is called. The send_in() function receives the return value (the size of the data sent), calculates the size of the data to be sent, and calls the send_in() function again if there is any data left to be sent.
Category	API Function

Function Name	u_int send_in(SOCKET s, u_char * buf, u_int len)
Arguments	s: Channel number buf: Pointer indicating the data to be sent len: Size of the data to be sent
Return value	Sent data size
Description	Internal function for sending TCP data. Called by the send() function for TCP transmission. It first calculates the free transmit buffer size and compares it with the size of the data to be transmitted to determine the transmission size. After calculating the data size, it copies data from TX_WR_PTR. It waits if there is a previous send command in process. When the send command is cleared, it updates the TX_WR_PTR up to the size to be transmitted and performs the send command.
Category	Internal Function

Function Name	u_int recv(SOCKET s, u_char * buf, u_int len)
Arguments	s: Channel number buf: Pointer where the data to be received is copied len: Size of the data to be received
Return value	Received data size
Description	TCP data receiving function. The recv() function is an application I/F function. It continues to wait for as much data as the application wants to receive.
Category	API Function

Function Name	u_int sendto(SOCKET s, const u_char * buf, u_int len, u_char * addr, u_int port)
Arguments	s: Channel number buf: Pointer indicating the data to send len: Size of the data to send addr: Destination IP address
Return value	Sent data size
Description	UDP data sending function. Composed of the sendto()and sendto_in() functions. The send() function is an application I/F function. It continues to call the send_in() function to complete the sending of the data up to the size of the data to be sent when the application is called. Unlike TCP transmission, it designates the destination address and the port.
Category	API Function

Function Name	**u_int sendto_in(SOCKET s, const u_char * buf, u_int len)**
Arguments	s: Channel number buf: Pointer indicating the data to send len: Size of the data to send
Return value	Sent data size
Description	UDP data sending function. An internal function that is the same as the send_in() function of the TCP.
Category	Internal Function

Function Name	**u_int recvfrom(SOCKET s, u_char * buf, u_int len, u_char * addr, u_int * port)**
Arguments	s: Channel number buf: Pointer where the data to be received is copied len: Size of the data to be received addr: Peer IP address for receiving port: Peer port number for sending
Return value	Received data size
Description	UDP data receiving function. Function for receiving UDP and IP layer RAW mode data, and handling the data header.
Category	API Function

Function Name	**char close(SOCKET s)**
Arguments	s: Channel number
Return value	1
Description	Channel closing function. Function for closing the connection of the designated channel.
Category	API Function

Function Name	u_int select(SOCKET s, u_char func)
Arguments	s: Channel number func: SEL_CONTROL(0x00) -> return socket status SEL_SEND(0x01) -> return free transmit buffer size SEL_RECV(0x02) -> return data size in receive buffer
Return value	Socket status or free transmit buffer size or received data size
Description	Function handling the channel socket information.
Category	API Function

Function Name	u_int read_data(SOCKET s, u_char * src, u_char * dst, u_int len)
Arguments	s: Channel number src: Receive buffer pointer of the W3100A dst: System buffer pointer len: Data size to be copied
Return value	Copied data size
Description	Copies the receive buffer data of the W3100A to the system buffer. It is called from the recv()or recvfrom() function.
Category	Internal Function

Function Name	u_int write_data(SOCKET s, u_char * src, u_char * dst, u_int len)
Arguments	s: Channel number src: System buffer pointer dst: Transmit buffer pointer of the W3100A len: Data size to be copied
Return value	Copied data size
Description	Copies the system buffer data to the transmit buffer of the W3100A. It is called from the send_in()or sendto_in() function.
Category	Internal Function

Function Name	void wait_10ms(int cnt)
Arguments	cnt: count
Return value	None
Description	Designates the delay. Waits for 10 milliseconds.
Category	Internal Function

Function Name	void wait_1ms(int cnt)
Arguments	cnt: count
Return value	None
Description	Designates the delay. Waits for 1 millisecond.
Category	Internal Function

Function Name	void wait_1us(int cnt)
Arguments	cnt: count
Return value	None
Description	Designates the delay. Waits for 1 millisecond.
Category	Internal Function