ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Schneider Electric / AEG / Modicon AT-984* at our website: ***Click HERE***

Chapter 7
Installing the OS/2 Files

- Installing the Driver and Link Files
- Installing the Remaining Files

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.1 Installing the Driver and Link Files

Before installing your files, review the information file READ.ME on your disks. This file lists each of your software programs and describes any recent updates. The text file MPHOS2.DOC contains a detailed description of your driver.

Installation of the device driver and dynamic link library is a three–step process:

- Copying the driver and dynamic link library files to the host computer's hard disk
- Editing the host computer's CONFIG.SYS file
- Initializing the host computer to recognize the new CONFIG.SYS configuration.

7.1.1 Copying MPHOST.SYS, MPHTICK.EXE, and SA85.DLL

MPHOST.SYS is the OS/2 driver for the single–cable or dual–cable SA85, SM85, AT–984, and MC–984. MPHTICK.EXE is the OS/2 driver application program. SA85.DLL is the Modicon NetBIOS dynamic link library file.

Use the OS/2 `COPY` command to copy the driver and dynamic link library files to your hard disk. Here are examples for copying the files to the C: OS/2 directories:

```
COPY MPHOST.SYS C:\OS2
COPY MPHTICK.EXE C:\OS2
COPY SA85.DLL C:\OS2\DLL
```

7.1.2 Editing CONFIG.SYS

Use your OS/2 system editor to add a DEVICE and RUN line for each installed device. Modify the LIBPATH environment to contain the path of the SA85.DLL file. A sample CONFIG.SYS file is provided on your disk. Here are examples:

```
DEVICE=C:\OS2\MPHOST.SYS /MC000 /N1
LIBPATH=C:\OS2\DLL
RUN=C:\OS2\MPHTICK.EXE /N1
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following parameters should also be present in CONFIG.SYS:

```
threads=128
memman=SWAP,MOVE
maxwait=3
timeslice=32,100
```

The DEVICE command line options are:

Switch	Meaning	Example	Default
/Mnnnn	Selects the memory window base address	/MD080 = address D0800	D0000
/Nn	Selects a specific device within the host	/N1 = device 1	device 0
/B	Blocks Board Reset and SA85OFF commands	/B = do not allow commands	allowed

/M
Use this argument to specify the same address that was set with the board's memory window base address DIP switches (SA85, AT−984) or software setup configuration (SM85, MC−984). Only four hexadecimal digits are used after the /M argument (e.g., /MD080) −− the rightmost 0 in the address is assumed.

/N
Use this argument to specify a unique Modicon device number within the host. Up to two devices can installed in the host, numbered 0 and 1. This argument identifies each board to the ncb_open() function in NetBIOS. If a value is not specified, the device driver defaults to device number 0.

/B
Use this argument to prevent Board Reset or SA85OFF commands in your application from resetting the board. If the command is not used, these commands will cause the board to initialize and start running its diagnostics.

7.1.3 Initializing the Host Computer

After you have installed the required device drivers and edited CONFIG.SYS, select 'Shutdown . . .' from the desktop window's 'Desktop' menu. Follow the screen instructions for shutdown. This will allow the host to recognize the CONFIG.SYS file changes.

If the host based device's switch settings, software setup, or device driver are incorrect, the host computer may hang. If this happens, recheck the installation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2 Installing the Remaining Files

7.2.1 Installing the Diagnostic and Sample Programs

The Network Diagnostic Utility program MBPSTAT.EXE is contained on your disk. You also have a set of sample programs that illustrate how to access controller registers and the network's global data.

Copy the diagnostic and sample program files to the directory you will want to use for exercising, testing, and debugging your Modbus Plus networking application. Descriptions of the programs are provided in the appendixes of this guide.

7.2.2 Installing the Headers, Source, and Library Files

The header files NETLIB.H and NETBIOS.H, source code file NETLIB.C (with your NetBIOS functions), and library file NETLIB.LIB are contained on your disk.

Refer to the manuals for the compiler you will use for your application program. Copy NETLIB.H and NETBIOS.H to your compiler's 'include' directory (typically \INCLUDE). Copy NETLIB.C to your compiler's working directory (typically \BIN). Copy NETLIB.LIB to the 'library' directory (typically \LIB).

NETLIB.LIB was compiled with Borland C version 3.1, using the Large model. You can link this file as required into your application program.

7.2.3 What to Do Next

After completing the software installation, your host based device is ready to be accessed through NetBIOS functions that you can program in your application. Descriptions of the NetBIOS functions, header files, and library are provided in the appendixes of this guide.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix A.
Using NetBIOS Functions

- The Modbus Plus NetBIOS Environment
- A Summary of Commands
- Return Codes
- ncb_cancel()
- ncb_close()
- ncb_open()
- ncb_receive()
- ncb_receive_datagram()
- ncb_receive_wait()
- ncb_reset()
- ncb_sa85off()
- ncb_send()
- ncb_send_datagram()
- ncb_set_slave_login()
- ncb_set_sw_interrupt()
- ncb_status()

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.1 The Modbus Plus NetBIOS Environment

NetBIOS is a C language software interface developed by IBM for PC networking. Modicon has adapted many of the NetBIOS functions to enable them to be used directly and efficiently in Modbus Plus networking. The applications designer who is familiar with standard NetBIOS functions will find the Modicon implementation easy to understand and apply.

Functions are provided for: initializing the local host based device; opening paths between the device and remote nodes; sending and receiving data or global data; receiving statistics from remote nodes; and retrieving the local device's statistics. The host can construct Modbus messages in buffers and send them to nodes. Incoming Modbus messages can be stored for use in the host application.

The syntax, arguments, and return values for these functions are described in this appendix. When using these functions, the designer should note their differences from the standard NetBIOS functions.

Before coding your application, you might want to examine the source code file netlib.c to view the NetBIOS function definitions. This file is on your disk. Also, you'll find a set of sample programs on your disk. You can examine them to see the use of these functions in actual Modbus Plus applications. You might want to printout a hard copy of the files to use as a reference in coding your application.

A.1.1 The Network Control Block

The data structure manipulated by the NetBIOS commands is the Network Control Block (NCB). As a NetBIOS command executes, all data is passed in the NCB, including pointers to external buffers.

The NCB is 64 bytes long and contains 14 fields of information. The fields are described on the next page. The columns are:

- Offset—the address of the field within the NCB (in hexadecimal)
- Size—the size in bytes of the field (in decimal)
- Field Name—the variable name of the field
- Field Description—the type of function or service performed by the field.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

NCB fields are defined in the header file netbios.h. A source code listing of the file is provided in Appendix C.

Offset (Hex)	Size (Bytes)	Field Name	Field Description
0	1	NCB_COMMAND	NetBIOS command code—high order bit set indicates no-wait (see netbios.h).
1	1	NCB_RETCODE	Completion result. - A value of zero if no error. See the list of return codes in this Appendix.
2	1	NCB_LSN	Not used and not supported.
3	1	NCB_NUM	A value identifying the type of internal path being used by the command. The value is placed in the field when the NCB is opened: 0x01 ... 0x08 DM1 ... DM8 0x41 ... 0x48 DS1 ... DS8 0x81 ... 0x88: PM1 ... PM8 0xC1 ... 0xC8 PS1 ... PS8
4	4	NCB_BUFFER	Address of the message buffer
8	2	NCB_LENGTH	Length of the message buffer, in bytes
0A	16	NCB_CALLNAME	The first 7 bytes identify the Modbus Plus routing path being used by the command. The contents are placed in the field when the NCB is opened. The contents are: Bytes 0 and 1 'DM', 'DS', 'PM', or 'PS' Bytes 2 ... 6 Routing addresses (binary) Other bytes are not used and not supported.
1A	16	NCB_NAME	The user's name on the network
2A	1	NCB_RTO	Receive timeout value in 500 ms increments. Used with ncb_receive_wait() and ncb_receive_datagram()
2B	1	NCB_STO	Send timeout value in 500 ms increments. Used with ncb_send() and ncb_send_datagram()
2C	4	NCB_POST_ADDRESS	Not used and not supported
30	1	NCB_LANA_NUM	Modbus Plus adapter number—must be 0 or 1.
31	1	NCB_CMD_CPLT	Not used and not supported
32	14	NCB_RESERVE	Reserved field, used internally by NetBIOS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.2 A Summary of Commands

The Modbus Plus NetBIOS commands and their corresponding functions in the NETLIB library are listed below.

Each function except exec_netbios() first initializes fields of the NCB structure. The function then calls exec_netbios() which invokes the int86x interrupt.

Command	NETLIB Function	Purpose
RESET	ncb_reset()	Resets the host based device and runs diagnostics
--	ncb_sa85off()	Turns off the device driver
CANCEL	ncb_cancel()	Aborts a transaction
STATUS	ncb_status()	Returns adapter status
ADD_NAME	ncb_open()	Obtains a path
DELETE_NAME	ncb_close()	Releases a path
SEND	ncb_send()	Sends a packet
RECEIVE	ncb_receive()	Receives a packet, no wait for response
RECEIVE_WAIT	ncb_receive_wait()	Same as RECEIVE except that it waits until the response is received or a timeout occurs
SEND_DATAGRAM	ncb_send_datagram()	Sends global data
RECEIVE_DATAGRAM	ncb_receive_datagram()	Receives global data
--	ncb_set_slave_login()	Sets the login status on a slave path
--	ncb_set_sw_interrupt()	Sets the software interrupt vector
--	exec_netbios()	Executes a NetBIOS command through the int86x interrupt. This function is an entry point for the device driver. Do not use it in your application.

A.2.1 Path Requirements

Some functions require the existence of a path in the local host-based adapter before they can execute. Four types of paths are available: DM (Data Master), DS (Data Slave), PM (Program Master), and PS (Program Slave). Paths are defined and established by the ncb_open() function, and released by ncb_close(). Other functions do not require the existence of a path in the adapter. They can be executed at any time without regard to the existence - or nonexistence - of paths.

Each function's description in this Appendix shows the types of paths supported.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.2.2 Message Buffers

The host application passes a buffer pointer value to ncb_send(), ncb_receive(), or ncb_receive_wait() that points to a buffer for the data to be passed in the function. Buffers can be constructed with messages in Modbus format for transactions with Modicon PLCs.

A pointer is passed to ncb_send_datagram() or ncb_receive_datagram() to identify the buffer for global data to be passed in the function.

A.2.3 Function Definitions and Prototypes

The function definitions for the NETLIB library are contained in the file netlib.c. You may want to print a hard copy of this file to use as a reference in coding your application.

The function prototypes are in the header file netlib.h.

A.2.4 NCB Structure Definition and NetBIOS Command Codes

The NCB structure is defined in the header file netbios.h. This file also defines the NetBIOS command codes.

A.2.5 Function Return Codes

The NCB_RETCODE values are defined in the header file netbios.h. A description of return code values is provided on the following pages.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.3 Return Codes

A hex value is returned to the NCB_RETCODE field to show the completion status of a function. The header file netbios.h defines the codes.

Code	include Variable	Description
00	`#define ERR_success 0`	Normal completion—no error
01	`#define ERR_bad_buffer_length 1`	Illegal send or status buffer length—send command length > 512 bytes, or status command length < the minimum allowed
03	`#define ERR_invalid 3`	Invalid NetBIOS command
05	`#define ERR_timeout 5`	The timeout interval specified for a data send or receive command has expired; or an internal timer for a status command has expired
06	`#define ERR_buffer_too_small 6`	Receive buffer too small—the buffer size specified in NCB_LENGTH is not large enough to hold the receive data
08	`#define ERR_bad_session_num 8`	Invalid local session number in NCB_LSN
09	`#define ERR_no_RAM 9`	Out of resources – the network adapter does not have enough RAM to process the command
0A	`#define ERR_session_closed 0xa`	Session has been terminated by the remote node
0B	`#define ERR_cancel 0xb`	Command cancelled—execution of NCB commands aborted by the ncb_cancel() command
0D	`#define ERR_dup_local_name 0xd`	Duplicate local name—an ncb_open() command has specified a name that already exists
0E	`#define ERR_name_table_full 0xe`	The local name table is full
0F	`#define ERR_active_session 0xf`	An ncb_close() command has been executed, but the name has active transactions; the name will not be deleted until all its sessions are closed
11	`#define ERR_sess_table_full 0x11`	The local session table is full
12	`#define ERR_no_listen 0x12`	Routing failure; remote node is not responding to the transaction
13	`#define ERR_bad_name_num 0x13`	Bad value in NCB_NUM field
14	`#define ERR_no_answer 0x14`	No answer to CALL, or no such remote
15	`#define ERR_no_local_name 0x15`	Invalid name entry in local name table
16	`#define ERR_duplicate_name 0x16`	Name already exists elsewhere on the network
17	`#define ERR_bad_delete 0x17`	Name has been incorrectly deleted
18	`#define ERR_abnormal_end 0x18`	Session terminated abnormally—connection to remote node terminated
19	`#define ERR_name_error 0x19`	Name conflict—two nodes using the same name have been detected
1A	`#define ERR_bad_packet 0x1a`	Bad NetBIOS packet on network
21	`#define ERR_card_busy 0x21`	Adapter busy—command cannot execute
22	`#define ERR_too_many_cmds 0x22`	Too many commands queued
23	`#define ERR_bad_card_num 0x23`	Invalid value for NCB_LANA_NUM—must be 0 or 1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

24	`#define ERR_cancel_done 0x24`	Command completed before ncb_cancel() executed; code returned in ncb_cancel() when previous command completed normally
26	`#define ERR_no_cancel 0x26`	Invalid ncb_cancel() command—target NCB cannot be found
40 ... FE		Hardware errors
FF	`#define ERR_busy 0xff`	Still processing command

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.4 ncb_cancel( )

NetBIOS Command
CANCEL

Description
Cancels and aborts a transaction before it has completed. The function does not clear an existing path—if the path is to be closed, issue a call to ncb_close() after completion of ncb_cancel().

Path Required
Requires the existence of a DM, DS, PM, or PS path.

Syntax

```
int ncb_cancel(*ncbp)
NCB *ncbp ;
```

Argument	
ncbp	A pointer to the NCB to be canceled.

Return Value	
ERR_success	Always

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.5 ncb_close( )

NetBIOS Command
DELETE_NAME

Description
Closes a transaction path that was opened with ncb_open(). If a transaction is pending on the path, the transaction is aborted.

Path Required
Requires the existence of a DM, DS, PM, or PS path.

Syntax

```
int ncb_close(*ncbp)
NCB *ncbp ;
```

Argument

ncbp	A pointer to the NCB returned by the previous ncb_open().

Return Values

ERR_success	Normal completion
ERR_bad_name_num	Invalid path

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.6 ncb_open( )

NetBIOS Command
ADD_NAME

Description
Opens a transaction path in the local host based device and designates a node with which you wish to communicate. You must call this function before you can communicate with another node. Pass an argument to the function that points to a string containing the path type and routing information. The path types are:

- DM—Data master path
- PM—Program master path
- DS—Data slave path
- PS—Program slave path

The string format is:

"PP.R1.R2.R3.R4.R5"

where:

- PP is the path type in the host based device: DM, PM, DS, or PS
- R1 ... R5 represent each level of Modbus Plus routing to the destination node

A period (.) character separates the fields within the string. You must zero-fill unused fields in the string, as in the examples below.

Example
Modbus Plus routing paths are described fully in Chapter 1. Here are two examples:

- Routing to a network adapter: DM.9.2.0.0.0 opens a Data Master path to task (slave path) 2 in the network adapter at node address 9.
- Routing to a controller: DM.8.3.0.0.0 opens a Data Master path to the controller at node 3 on the network that is accessed through the Bridge Plus device at node 8 on the local network.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Sample Call

```
ncb_open("DM.8.3.0.0.0", 0) ;
```

where

- `"DM.8.3.0.0.0"` is a string defining the path type and routing.
- `0` selects host based device 0.

Pointer to NCB

The function returns a pointer to an NCB, or a NULL on an error. The pointer must be passed to the following commands:

ncb_send()	ncb_receive_wait()	ncb_close()
ncb_receive()	ncb_set_slave_login()	ncb_cancel()

Path Required

Not applicable. The function establishes a new DM, DS, PM, or PS path.

Syntax

```
NCB *ncb_open(*name, lan)
char *name ;
int lan ;
```

Arguments

`name`	A pointer to a string containing the path type and routing.
`lan`	Selects the host based device (0 or 1).

Return Values

pointer to an NCB	See **Pointer to NCB** above
NULL	If an error has occurred

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.7 ncb_receive( )

NetBIOS Command
RECEIVE

Description
Receives a message on the specified path. If the response is available, the function returns with the response. If the response is not available, it returns immediately with ERR_timeout status.

Use this function instead of ncb_receive_wait() to enable better throughput in a multitasking environment.

Path Required
Requires the existence of a DM, DS, PM, or PS path.

Syntax

```
int ncb_receive(*ncbp, *buffer)
NCB *ncbp ;
char *buffer ;
```

Arguments

ncbp	A pointer to the NCB returned by the previous ncb_open().
buffer	A a pointer to the buffer for the received message.

Return Values

ERR_success	Good
ERR_bad_session_num	NCB_NUM == 0
ERR_bad_name_num	Invalid path
ERR_timeout	No response available
ERR_invalid	No response pending
ERR_busy	Bad path (NCB_NUM)
ERR_invalid	Returned length of 0 (NCB_LENGTH)
ERR_no_listen	Node not present

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.8 ncb_receive_datagram( )

NetBIOS Command
RECEIVE_DATAGRAM

Description
Receives a specified quantity of global data into a buffer. The user must allocate an NCB for the function that is separate from NCBs allocated by ncb_open(). After the return, NCB_LENGTH contains the count of global data words received.

Path Required
Not applicable. The function does not require a path.

Syntax

```
int ncb_receive_datagram(*ncbp, node, *buffer, timeout, adaptno)
NCB *ncbp ;
int node ;
char *buffer ;
unsigned char timeout ;
int adaptno ;
```

Arguments

ncbp	A pointer to user–allocated NCB (should be empty).
node	The node from which to receive global data (range: 1 ... 64).
buffer	A pointer to the buffer that will contain the received data.
timeout	The timeout value in units of 0.5 s.
adaptno	The host device adapter number—must be 0 or 1.

Return Values

ERR_success	Good
ERR_card_busy	Node not found
ERR_no_answer	No global data present

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.9 ncb_receive_wait( )

NetBIOS Command
RECEIVE_WAIT

Description
Receives a message on the specified path. Unlike ncb_receive, this function waits until the response comes in or until the timeout occurs before it returns.

Path Required
Requires the existence of a DM, DS, PM, or PS path.

Syntax
```
int ncb_receive_wait(*ncbp, *buffer, timeout)
NCB *ncbp ;
char *buffer ;
unsigned char timeout ;
```

Arguments

ncbp	A pointer to the NCB returned by the previous ncb_open().
buffer	A a pointer to the buffer for the received message.
timeout	The timeout value in units of 0.5 s.

Return Values

ERR_success	Good
ERR_bad_session_num	NCB_NUM == 0
ERR_bad_name_num	Invalid path
ERR_timeout	No response available
ERR_invalid	No response pending
ERR_busy	Bad path (NCB_NUM)
ERR_invalid	Returned length of 0 (NCB_LENGTH)
ERR_no_listen	Node not present

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.10 ncb_reset( )

NetBIOS Command
RESET

Description
Resets the network adapter, runs its diagnostics, and returns it to a known initial state. The function may require up to 100 ms to execute.

This function can be either allowed or blocked by the device driver through the use of driver command line parameters.

The function will be blocked if you are running under Microsoft Windows 3.x or under DesqVIEW. If the function is called, it will have no effect unless the driver has crashed and requires a reset.

Path Required
Not applicable. The function does not require a path.

Syntax

```
int ncb_reset(adaptno)
int adaptno ;
```

Argument

`adaptno`	The number of the adapter to be reset—must be either 0 or 1.

Return Values

`ERR_success`	Reset occurred normally
`ERR_invalid`	Bad adapter number
`0x80`	Diagnostic failed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.11 ncb_sa85off( )

NetBIOS Command
None

Description
Turns off the device driver. Use this function when your application no longer needs the host based device, and you want to run some other type of program such as a terminal emulation program. It turns off the polling of the host based device by the device driver. Use the ncb_reset() function to reset the device and restore its operation.

This function can be either allowed or blocked by the host based device driver through the use of driver command line parameters.

The function will be blocked if you are running under Microsoft Windows 3.x or under DesqVIEW. If the function is called, it will have no effect.

Path Required
Not applicable. The function does not require a path.

Syntax
```
int ncb_sa85off(adaptno)
int adaptno ;
```

Argument

adaptno	The number of the adapter to turn off. Must be either 0 or 1.

Return Values

ERR_success	Normal completion
ERR_bad_card_num	Bad adapter number

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.12 ncb_send( )

NetBIOS Command
SEND

Description
Sends a message on the selected path. For a master path, you must issue ncb_send() before ncb_receive() or an error will be returned. For a slave path, you must issue ncb_receive() before ncb_send() or an error is returned.

Path Required
Requires the existence of a DM, DS, PM, or PS path.

Syntax
```
int ncb_send(*ncbp, length, *buffer, timeout)
NCB *ncbp ;
int length ;
char *buffer ;
unsigned char timeout ;
```

Arguments

ncbp	A pointer to the NCB returned by the previous ncb_open().
length	The length of the message buffer contents in bytes.
buffer	A pointer to the buffer containing the message to be sent.
timeout	The timeout value in units of 0.5 s.

Return Values

ERR_success	Good
ERR_bad_session_num	NCB_NUM == 0
ERR_invalid	If response is still pending on the master path, or if no previous receive has been issued on the slave path

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.13 ncb_send_datagram( )

NetBIOS Command
SEND_DATAGRAM

Description
Sends a message containing a specified quantity of global data. The user must allocate an NCB for the function that is separate from NCBs allocated by ncb_open().

Path Required
Not applicable. The function does not require a path.

Syntax

```
int ncb_send_datagram(*ncbp, length, *buffer, timeout, adaptno)
NCB *ncbp ;
int length ;
char *buffer ;
unsigned char timeout ;
int adaptno ;
```

Arguments

ncbp	A pointer to user–allocated NCB (should be empty).
length	The length of the message buffer contents in bytes.
buffer	A pointer to the buffer containing the message to be sent.
timeout	The timeout value in units of 0.5 s.
adaptno	The host device adapter number—must be 0 or 1.

Return Values

ERR_success	Good
ERR_card_busy	Adapter not found or is initialized
ERR_bad_card_num	Bad adapter number
ERR_bad_buffer_length	Buffer length > 64

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.14 ncb_set_slave_login( )

NetBIOS Command:
None

Description
Sets or clears login status on a Program Slave path in the local adapter to the local host application. Once the path is established it can be used by a login command and subsequent programming commands from a remote application.

Path Required
Requires the existence of a PS path.

Syntax

```
int ncb_set_slave_login(*ncbp, login_status)
NCB *ncbp ;
unsigned char login_status ;
```

Arguments

ncbp	A pointer to the NCB returned by the previous ncb_open().
login_status	The desired login status: 1 sets login. 0 clears login.

Return Values

ERR_success	Good
ERR_busy	Error or adapter busy
ERR_bad_name_num	Bad NCB_NUM field
ERR_no_listen	Path is inactive

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.15 ncb_set_sw_interrupt( )

NetBIOS Command
None

Description
Sets the software interrupt vector used to access the device driver to the value passed to the function.

Path Required
None

Syntax
```
int ncb_set_sw_interrupt(swInterrupt)
int swInterrupt ;
```

Argument

`swInterrupt`	The desired software interrupt vector value. Valid arguments are: 3...7, 10...FF. The vector value must match the value specified in the CONFIG.SYS file.

Return Values

`ERR_success`	Good
`ERR_invalid`	Did not affect the current interrupt vector setting

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.16 ncb_status( )

NetBIOS Command
STATUS

Description
Returns the local host based device configuration. The user must allocate an NCB for the function that is separate from NCBs allocated by ncb_open().

The function will receive a total of 110 bytes of status. The message buffer pointer in NCB_BUFFER must point to a valid buffer area of at least 110 bytes in length.

The first two bytes received into the buffer will be a diagnostic pattern 0xAA, 0x55 (alternating binary 0 and 1 in each bit position). This verifies that each bit position can be set and reset.

Bytes 2 - 109 will contain the status information. This is the same status as is returned by Modbus function 8, subfunction 21 (Get/Clear Network Statistics). See the description of that function in Appendix F for a listing of status values. The ncb_status() function does not clear the statistics.

Path Required
Not applicable. The function does not require a path.

Syntax

```
int ncb_status(*ncbp, adaptno)
NCB *ncbp ;
int adaptno ;
```

Arguments

ncbp	A pointer to the user-allocated NCB.
adaptno	The host adapter device number—must be either 0 or 1.

Return Values

ERR_success	Good
ERR_invalid	Diagnostic failed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix B.
Modifying the 5C Interrupt

□ Modifying the 5C Interrupt

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Modifying the 5C Interrupt

The software interrupt number to be used in communicating to the device driver (DOS applications only) is contained in the variable sw_interrupt, which is declared in the NETLIB.C code module.

The interrupt number is not applicable when using OS/2.

B.1.1 Interrupts in Your Application

To change the software interrupt in your application, first refer to the chapter in this guide for installing the DOS software, and modify the interrupt parameter in the DEVICE line in your CONFIG.SYS file. Then have your application change the contents of the sw_interrupt variable to contain the new software interrupt number.

The following code fragment illustrates how to do this:

```
/* main.c - arg1 is the software interrupt number
usage: main /s5C */

extern int sw_interrupt;

int main( int argc, char *argv[] )
{
/* read command line parameter - if not assigned,
assign default */

if(sscanf( argv[1], "/S%x", &sw_interrupt) != 1 )
    sw_interrupt = 0x5c ;
}
```

Using Windows DLLs: Note that the above code will not work with applications that use Windows DLLs. In these applications, use the ncb_set_sw_interrupt() function described below.

You can also modify the interrupt within your application by a call to the ncb_set_sw_interrupt() function, described in Appendix A. You can find an example of the use of this function in the sample program BDRESET.C.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1.2 Interrupts in MBPSTAT

As supplied, your Network Diagnostic Utility program MBPSTAT accepts the /S command argument for the software interrupt. When calling MBPSTAT, pass it the argument you used in the DEVICE line in your CONFIG.SYS file.

B.1.3 Interrupts in Your Sample Programs

As supplied, your host based device board reset programs SA85OFF and BDRESET accept the /S command argument for the software interrupt. When calling either of these programs, pass it the argument you used in the DEVICE line in your CONFIG.SYS file.

Your other sample programs are supplied as coding examples, rather than as standalone programs capable of being run directly on your network configuration. Before running any program, refer to the program descriptions in Appendix E, including the caution about modifying the interrupt and node addresses in the programs. You can use the code fragment on the previous page as an example for editing the source code to use a different interrupt.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix C.
Include Files

- netbios.h
- netlib.h

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

C.1 netbios.h

```
/*=============================================================================
 *      MODICON, Inc., 1 High Street, North Andover, MA 01845, USA
 *      Copyright (c) $Date: 1994/05/02 14:51:32 $ - All rights reserved.
 *
 *      No part of this document may be reproduced in any form
 *      without the express written consent of MODICON, Inc.
 *
 *=fdoc========================================================================
 *
 * File:        netbios.h
 *
 *
 * Notes:       Include file for NetBIOS calls and definitions
 *
 * Author:
 *
 * RCSID:
 *
 * RCS:
 * $Revision: 3.33 $
 * $State: R3_4B1 $
 * $Locker:  $
 *
 * Revision History Log:
 *
 * Revision 3.32  03/11/94
 * Baseline
 *
 *=fdoc========================================================================*/
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP