

- fonts and grids 8-19
- getcontrast() function 8-33
- getfont() function 8-30
- getfontinfo() function 8-31
- getgrid() function 8-31
- gotoxy() function 8-29
- graphical image, writing 8-33
- graphics and characters, creating 8-19
- grids 8-19
- non-ASCII fonts 8-23
- open() function 8-26
- putpixelcol() function 8-33
- raw font 8-23
- raw mode 8-18
- raw mode character 8-23
- read() function 8-27
- related functions 8-34
- resetdisplay() function 8-31
- retrieving font information E-2
- selecting a new font 8-25
- setcontrast() function 8-32
- setfont() function 8-30
- translation tables 8-24
- window() function 8-28
- windows, using display 8-19, 8-44
- write() function 8-27
- display, programming notes, tips 3-14
- display, segment-type:
 - close() function 8-13
 - display related functions 8-14
 - open() function 8-12
 - programming example 8-15
 - read() function 8-13
 - windows, using display 8-14
 - write() function 8-12
- DISPLAYC 8-15

display:

- pixel-type, programming 8-17 – 8-48
- segment-type, programming 8-11 – 8-16
- displaying four lines 8-37
- displaying three lines 8-36
- displaying two lines 8-35
- document conventions 1-6
- double-wide characters, pixel-type display 8-25
- download control 4-10
- downloading 2-3
 - application-initiated 2-4
 - terminal-to-terminal 2-3
- DSP.C 11-128
- DSP2HEX.C 11-120
- DTR signals, terminal differences 9-1

E

- 8-bit character codes 8-21
- enable call progress modem command 9-48
- environment variables, LAN 10-15
- environment variables, reading CONFIG.SYS 4-11
- environment variables, writing to CONFIG.SYS 4-11
- EPROM version 11-129
- EPROM version 9 or earlier, attention users 1-5
- EPROM.C 11-129
- errno, using 3-5
- error codes 3-6
- error messages, terminal 2-18
- executable file 6-2

F

- _free() 11-35 – 11-36, F-12
- FDF
 - see font definition files
- features, OMNI 300 Series terminals 1-3

file functions:

- close() 6-19
- close_all() 6-19
- delete() 6-17
- dir_get_first() 6-34
- dir_get_next() 6-34
- dir_get_sizes() 6-33
- getkey() 6-23
- insert() 6-16
- ioctl() 6-18
- lseek() 6-15
- open() 6-11
- putkey() 6-23
- read() 6-13
- remove() 6-35
- write() 6-14

file handles 6-2

file manager 6-7

file naming 6-2

file space, calculating available 5-5

file space, maximizing available 5-1

file space, minimizing use of 3-13

file storage 6-2

file system, stored in non-volatile memory 3-8

file types and access methods 6-3 – 6-8

FILE.C 11-15

file_copy() 11-37 – 11-38

FILE_COPY.C 11-38

FILEOPS.C 11-55

files operations:

- dir_get_first() 11-28
- dir_get_next() 11-29
- dir_get_sizes() 11-30

files:

- <config.h> 8-2
- <io.h> 8-2
- closing 6-19
- conventions 6-2
- creating new 6-11
- date and time stamp 6-19
- deleting data 6-17
- design considerations 3-13
- directory functions 6-33
- generic 6-3
- handle 6-2
- handles, maximum 6-11
- inserting data 6-16
- keyed 6-21
- keyed, tips on using 3-13
- management 6-1
- naming 6-2
- new 6-11
- opening 6-10
- opening for write access 6-11
- positioning 6-12, 6-15
- reading from 6-13
- storage 6-2
- types and access methods 6-3 – 6-8
- writing to 6-14

- FILESYS.C 6-20
- font (pixel-type display), defined 8-17
- font basics 8-17
- font codes, character 8-24
- font definition file version number 8-21
- font definition file, default 8-22
- font definition files 8-21
- font page 8-17
- font page, selecting 8-23
- font pages, display characters from different 8-38

font ROM chip 8-17
font ROM page 8-21
font set tables E-3 – E-6
font set:
 Hong Kong E-5
 People's Republic of China E-6
 Saudi Arabia E-4
 Taiwan E-5
 Thailand font set "A" E-3
 Thailand font set "B" E-4
font, changing with CONFIG.SYS 4-6
FONT.C 11-41
FontDesigner 8-19
FONTINFO.C 11-43
fonts and grids, pixel-type display 8-19
fonts, pixel-type display 8-19
fonts:
 font 11A1112E E-7
 font 11A2112E E-7
 font 11A3112E E-8
 font 12N2111E E-8
 font 12N2112E E-8
 font 12N3111E E-8
 font 13N4111E E-9
 font 13N4121E E-9
 font 14N4112E E-3, E-9
 font 14N4122E E-10
 font 14N4132E E-10
 font 15N4111E E-10
 font 15N4122E E-11
 font 15N4131E E-11
 font 16N3112E E-11
 font 16N3122E E-12
 font 17N4112E E-12
 font 18N4112E E-12
font set E-6

font set 1112A11E E-3
font set 1124A11E E-4
font set 1134A11E E-4
font set 1143A11E E-5
font set 1153A11E E-5
font set 1162A11E E-6
400 series LAN protocols F-13
function keys, terminal 8-3, 8-5

G

get_env() 4-11, 11-40
GET_ENV.C 11-40
get_lan_config() 10-19, 11-47 – 11-49
getcontrast() 11-39
getenv() 4-11
getfont() 11-41
getfontinfo() 11-42 – 11-43, E-2
getgrid() 11-44
GETGRID.C 11-44
getkey() 6-23, 11-45 – 11-46
GETLCFG.C 11-49
getscrollmode() 11-50 – 11-51
global data, initializing 5-9
global variables 3-13
go back on-line modem command 9-47
go off hook, modem command (dial command) 9-46
gotoxy() 11-52 – 11-53
grid, display, changing with CONFIG.SYS 4-6
grid, pixel-type display, defined 8-19
GRID_2x18 8-20
GRID_3x18 8-20
GRID_4x25 8-20
guard function:
 creating 7-8 – 7-13
 defined 7-4

guard list management example 7-5
guard list, defined 7-5
guard tones, modem, CCITT, enable 9-49
guard, defined 7-1
guards 7-4 – 7-7
GUARDS.C 7-17

H

H modem command 9-47
handshake, eliminate modem off-hook command 9-51
hang up modem command 9-47
hash character
 see pound character
Hayes-compatible modem control 9-43
header files, listing 3-2 – 3-3
heap allocation error, trap 7-2
HELLO2.C 8-35
HELLO3.C 8-36
HELLO4.C 8-37
HEX2DSP.C 11-122
hexidecimal to ASCII conversion 11-122
hide mask 7-4
Hong Kong font set E-5
hyphen (-) modem command 9-46

I

ignore flag, exception handling 7-4
illegal EM instruction error, trap 7-2
include files 3-2
INFO_KEY.C 11-131
initialization files, avoiding 3-14
initialization, modem 9-52
insert() 6-16, 11-54 – 11-55
insert_cv1r() 6-16, 11-56 – 11-57

insert_vlr() 6-16, 11-58 – 11-59
insline() 11-60
INSLINEC 11-60
INT2.C 11-136
ioctl() function 11-61
ioctl():
 bar code reader 8-62
 beeper 8-49
 card reader 8-56
 clock 8-67
 files 6-18
 keyboard 8-7
 LAN 10-24
 modem 9-32
 PIN Pad 9-23
 serial port 9-14
 ioctlc() 10-30, 11-61

K

key codes 8-3 – 8-4
KEYC 11-46
KEY_NUM.C 11-138
KEY_TXT.C 11-140
keyboard buffer size 8-3
keyboard constants 8-4
keyboard functions:
 close() 8-8
 ioctl() 8-7
 open() 8-6
 read() 8-6
 write() 8-7
keyboard traps 7-2
KEYBOARD.C 8-9

keyboard:
 device name 8-2
 event trap 8-7
 programmable keys 8-5
 programming example 8-9
 KEYDFILE.C 6-22
keyed files:
 reading/writing 6-21
 ZONTALK restrictions 6-21
 getKey() 11-45 – 11-46
 putkey() 11-77 – 11-78
 tips 3-13
keypad:
 core 4x4 8-4
 key coding 8-4
 see also keyboard, programmable keys

L

LAN baud rate, setting in CONFIG.SYS 4-7
LAN download all flag, setting in CONFIG.SYS 4-9
LAN download server, setting in CONFIG.SYS 4-7
LAN download type, setting in CONFIG.SYS 4-7
LAN downloading, defined 2-4
LAN downloading:
 application-initiated download 10-39
 download all terminals 10-44
 empty terminal at power up 10-37, 10-41
 NO_Dial Flag 10-40
 server-initiated downloads 10-40
 terminal in system mode 10-38
 terminal running application 10-44
 LAN environment variables, read/write 4-11

LAN functions: 10-16 – 10-30

close() 10-30

get_lan_config() 10-19

ioctl() 10-24

ioctlc() 10-30

open() 10-16

put_lan_config() 10-17

read() 10-21

write() 10-22

LAN high address, setting in CONFIG.SYS 4-8

LAN terminal address, setting in CONFIG.SYS 4-6

LAN timing window, setting in CONFIG.SYS 4-8

LAN:

application message type in packet header 10-4

buffer management 10-12

app_type 10-4

BUS length maximum 10-2

data exchange 10-4 – 10-11

data length parameter in packet header 10-4

data packet fields 10-5

data packet, defined 10-4

destination address in packet header 10-4

dg_length 10-4

DIAG_REQ message 10-8

download message 10-8

download protocol and packets 10-45

download request packet 10-46

download sequence numbers 10-49

dst_addr 10-4

error recovery 10-10

gateway terminal 10-1

hardware 10-1

I_M_HERE message 10-9

message type in packet header 10-4

NORM message 10-8

OSI architectural standard 10-2

packet CRC 10-4
 packet data with packet header 10-4
 protocol driver, defined 10-1
 protocol reset message 10-9
 remote diagnostics 10-13
 sequence number in packet header 10-4
 source address in packet header 10-4
 src_addr 10-4
 standard ZONTALK packets 10-48
 terminal installation procedures 10-3
 traps 7-3
 type parameter in packet header 10-4
 tx_seq 10-4
 line presence, check modem command 9-46
 literal strings, avoiding 5-8
 longjmp(), avoid with traps 7-11
 LRC generation 11-141 – 11-142
 lseek() 6-15, 6-18, 11-62 – 11-63

M

magnetic card reader, device name 8-2
 mask_and() 7-7, 11-64
 MASK_AND.C 11-64
 mask_clear() 7-7, 11-65
 MASK_CLR.C 11-65
 MASK_EMP.C 11-66
 mask_empty() 7-7, 11-66
 MASK_FIL.C 11-67
 mask_fill() 7-7, 11-67
 mask_in() 7-7, 11-68
 MASK_INC 11-68
 mask_of() 7-7, 11-69
 MASK_OF.C 11-69
 mask_or() 7-7, 11-70
 MASK_OR.C 11-70

masks, exception handling	7-4
MDMHOST.C	9-37
MDMVISA.C	9-40
memory size	11-145
memory utilization	5-1
memory, general considerations	5-10
memory:	
non-volatile	3-8, 5-2
volatile	3-8, 5-2
MOD_CK.C	11-142
modem commands, table	9-45
modem EPROM table	9-52
modem functions:	
close()	9-32
ioctl()	9-32
open()	9-29
read()	9-30
read_cmd()	9-31
write()	9-30
write_cmd()	9-32
modem interface, defined	9-27
modem port	9-27 – 9-64
modem:	
baud rate, setting	9-10
commands, Hayes	9-45
commands, off hook and dialing	9-50
defaults (in EPROM)	9-52
factory initialization	9-52
Hayes-compatible	9-43
ignored characters and commands	9-45
initialization with CONFIG.SYS	9-52
off-hook and dialing commands	9-50
protocol code	9-11
receiving responses	9-44
sending commands	9-43

MODEMCMD.C 9-42, 11-86
modes, communication 9-4 – 9-10

N

new font, selecting 8-25
non-banked memory
 see volatile memory
non-volatile memory 3-8
non-volatile memory management 5-3
non-volatile memory, defined 5-2
null pointer, warning about using zero value F-6
NULL-terminated strings, defined 3-4

O

.OUT file 3-8, 6-2
O modem command 9-47
O3X5LAN.C 10-50 – 10-61
O_APPEND 6-10, 6-12
O_CREAT 6-11
O_CREATO_CREAT 6-10
O_EXCL 6-10
O_RDONLY 6-10
O_RDWR 6-10
O_TRUNC_TRUNC 6-10
O_WRONLY 6-10
off-hook modem commands 9-50
off-hook, go off-hook modem command 9-46
OMNI 300 Series terminal, defined 1-1
OMNI 400 Series terminal 1-1
OMNI 400 series terminals, differences F-2
OMNI 460 integral printer
 commands D-6

OMNI 460 integral printer:
character map, country specific D-4
enhancements D-5
fonts D-3
graphics D-5
line spacing D-5
paper detection circuit D-5
print mechanism D-2
print motor timing test D-6
printable character set D-3 – D-5
open files, maximum number of 6-11
open() function 11-71 – 11-72
open():
bar code reader 8-60
beeper 8-49
card reader 8-54
clock 8-66
display 8-12
file attributes 6-10
files 6-10
LAN 10-16
modem 9-29
PIN Pad 9-21
serial port 9-12
open_codefile() 11-73 – 11-74
OPENCODE.C 11-74
operating system, TXO, defined 1-4
Opn_Blk structure 9-8

P

P command, modem off-hook 9-51
PACK4.C 11-144
packet mode, communications 9-5
packet, communications, maximum data 9-5
page and offset codes, font 8-21

paired files
 see keyed files

pause character, modem dialing character 9-51

pending events, exception handling 7-5

pending_traps() 7-14, 11-75

People's Republic of China font set E-6

period character:
 pixel-type display 8-18

segment-type display 8-11

PIN pad port functions:
 close() 9-23

 ioctl() 9-23

 open() 9-21

 read() 9-22

 write() 9-22

PIN pad port: 9-21 – 9-26

errors 9-24

Opn_Blk structure 9-25

resetting errors 9-24

status 9-24

PIN pad traps 7-3

PINPAD.C 9-25

PLATFORM.C 11-132

portability:
 barcode reader F-14

 beeper F-14

 byte order and word order F-1, F-7

 character sets F-4

 checking dsp_90.h header files F-5

 clear key trapping during svc_key functions F-11

 clock ticks F-10

 display functions F-5

 display source code F-3 – F-4

 exception handling and trap numbers F-10

 400 series LAN buffers F-12

 400 series LAN protocols F-13

implicit and explicit opens	F-9
keyboard functionality	F-14
LAN Differences	F-11
memory utilization	F-9
modem ioctl() function differences	F-13
multi-tasking	F-9
OMNI display types	F-1
pointer size	F-1, F-5 – F-6
SVC_WAIT()	F-13
pound character, in CONFIG.SYS	4-1
Printer controller	
functions	D-5
printer port	
<i>see serial port</i>	
program chaining	3-9, 5-9
program file	6-2
program space, 45Kb limit	3-13
programming requirements	1-4
protocol code, modem	9-11
protocol, communications, setting	9-11
PTID.C	11-133
pulse dial ratio, set modem command	9-49
pulse dialing, modem off-hook command	9-51
punctuation, source code	1-7
put_env()	4-11, 11-76
PUT_ENV.C	11-76
put_lan_config()	10-17, 11-79 – 11-80
putkey()	6-23, 11-77 – 11-78
PUTLCFG.C	11-80
PUTPIXELC	11-82
putpixelcol()	11-81 – 11-82
R	
RAM_SIZE.C	11-145
raw font, defined	8-23

raw mode character 8-23
raw mode characters, writing 8-40
raw mode, pixel-type display 8-18
RAW.C 8-42
read() function 11-83 – 11-84
read():
 bar code reader 8-61
 card reader 8-54
 clock 8-66
 display 8-13
 files 6-13
 keyboard 8-6
 LAN 10-21
 modem 9-30
 PIN Pad 9-22
 serial port 9-12
 read_cmd() 11-85 – 11-86
 modem 9-31
 read_cv1r() 6-13, 11-87 – 11-88
 read_reject_pkt() 11-89 – 11-90
 read_v1r() 6-13, 11-91 – 11-92
 READREJ.C 11-90
real time clock functions:
 close() 8-67
 ioctl() 8-67
 open() 8-66
 read() 8-66
 write() 8-66
real time clock:
 defined 8-65
 device name 8-2
 differences between OMNI 300 and
 OMNI 400 series terminal 8-65
 RECALL? prompt, SVC_STORE() 11-149 – 11-150
 receiving responses, modem 9-44
 reference manuals, OMNI 300 Series 1-3

reject queue, LAN, emptying 10-25
 reject queue, LAN, queueing 10-25
 reject queue, modem, reading 9-35
 remove() 6-35
 requirements, programming 1-4
 resetdisplay() 11-93
 RESETDSPC 11-93
 RS-232
 see serial port
 RS-485, LAN 10-2
 RS232CHR.C 9-17

S

7-bit character codes 8-21
 S modem command 9-47
 S registers, setting modem command 9-47
 Saudi Arabia font set E-4
 screen addressable keys 8-3
 SCROLL.C 11-51
 SDLC 9-8
 Hypercom 9-8
 SDLC protocol 9-4, 9-8
 SDLC, support on OMNI 300 series terminals 9-28
 secondary dial tone, wait for 9-52
 seek pointer 6-12
 SEEK_CUR 6-15
 seek_cv1r() 6-15, 11-94 – 11-95
 SEEK_END 6-15
 SEEK_SET 6-15
 seek_v1r() 6-15, 11-96 – 11-97
 seeking, files 6-12
 segment-type display
 see display, segment-type
 sending commands, modem 9-43
 serial communications 9-11

serial port 9-11 – 9-19

serial port functions:

- close() 9-13
- ioctl() 9-14
- open() 9-12
- read() 9-12
- write() 9-13

serial port:

- baud rate, setting 9-10
- constants 9-10
- errors 9-15
- Opn_Blk structure 9-17
- programming example 9-17
- resetting errors 9-15
- status 9-16

set Bell/CCITT mode modem command 9-46

set pulse dial ratio modem command 9-49

set S-register values modem command 9-47

set_env_buffer() 4-11, 11-100 – 11-101

setcontrast() 11-98 – 11-99

setfont() 8-25, 11-102 – 11-103

setjmp(), avoid with traps 7-11

setscrollmode() 11-104 – 11-105

setup, terminal 2-1

shared memory 5-3

shifted keys and key codes, terminal keyboard 8-4

slot number, exception handling, defined 7-4

soft keys, keyboard

- see keyboard, programmable keys

software delay, SVC_WAIT() function 11-157

sound() 8-50, 11-106 – 11-107

sound(), programming example 8-51

SOUND.C 11-107

source code, conventions 1-6

stack and heap, maximizing 5-1

stack overflow trap 7-2

stack size, increasing 4-9

Standard C for TXO 3-1

startup:

- delays 6-8
- display font upon power up 8-38
- rules 3-9
- sequence 2-5
- speeding up 4-5

static records 6-3

Status request packet D-7

Status response packet D-8

STDERR 8-2

STDIN 8-2

STDOUT 8-2, 8-12

STORE.C 11-150

strings:

- counted 3-4
- NULL-terminated 3-4
- SVC_ZINT() 11-153
- SVC_CHECKFILE() 11-108 – 11-109
- SVC_CHK_PASSWORD() 11-110
- SVC_CLOCK() 8-66, 11-111 – 11-112
- SVC_COUNT() 11-113 – 11-116
- SVC_CRC_CALC() 11-117 – 11-118
- SVC_DSP_2_HEX() 11-119 – 11-120
- SVC_GET_UNIT_ID() 11-121
- SVC_HEX_2_DSP() 11-122
- SVC_INFO_BAR() 11-123 – 11-124
- SVC_INFO_CARD() 11-125 – 11-126
- SVC_INFO_DSP() 11-127 – 11-128
- SVC_INFO_EPROM() 11-129
- SVC_INFO_KEY() 8-8, 11-130 – 11-131
- SVC_INFO_PLATFORM() 11-132
- SVC_INFO_PTID() 11-133
- SVC_INFO_TYPE() 11-134 – 11-135
- SVC_INT2() 11-136

SVC_KEY_NUM() 11-137 – 11-138
SVC_KEY_TXT() 11-139 – 11-140
SVC_MOD_CK() 11-141 – 11-142
SVC_PACK4() 11-143 – 11-144
SVC_RAM_SIZE() 11-145
SVC_RESTART() 11-146 – 11-148
SVC_STORE() 11-149 – 11-150
SVC_TICKS() 11-151 – 11-152
SVC_UNPK4() 11-154 – 11-155
SVC_VERSION() 11-156
SVC_WAIT() 11-157
SVC_ZONTALK(), LAN 10-40, 10-44, 11-160
SVC_ZONTALK(), modem 11-158
SVCLOCK.C 11-112
sync/async port 9-20
synchronous mode, described 9-28
system files, default 6-2
system mask, exception handling 7-8
system mode 2-23
system mode:
 functions by key 2-24
SYSTEM.BIN:
 default system file 6-2
 defined 5-2

T

T command, modem off-hook 9-51
TAIL_EM, with EPROM version 9 or earlier 1-5
Taiwan font set E-5
take mask 7-4
TCKMPC 11-162
TCL 1-4
terminal setup 2-1
terminal startup sequence 2-5
terminal-to-terminal download 2-3

terminal-to-terminal downloads, deleted data4-2

terminology used in this manual 1-2

Thailand font set "A" E-3

Thailand font set "B" E-4

tick_compare() 11-162

TICKS.C 11-152

tips:

- *LFP setting, no value enters system mode 10-16
- *LZF, using to download to all terminals 10-16
- getkey() and putkey() versus CVLR files 6-21
- file records consuming storage capacity 6-8
- file space, minimizing 3-13
- fonts, displaying characters from different 8-20
- function defaults to int when type long 3-2
- global variables 3-13
- guidelines 3-13 – 3-14
- initialization files, avoiding 3-14
- inserting and deleting file data 6-17
- keyed files and CONFIG.SYS 3-13
- keyed files, avoiding 6-7
- managing application data 5-4 – 5-8
- minimizing file size 6-1
- mixing fonts 8-20
- operating system buffers 3-14
- pixel-type display 8-35
- pixel-type display, clearing and resetting buffer 8-20
- preseving application between power cycles 3-8
- program space 3-13
- startup routine for always-open files 3-8
- writing guard functions 7-9
- tone dialing, modem off-hook command 9-51
- track data format, card reader 8-54
- transactions, processing time 3-14
- translation tables, pixel-type display 8-24
- trap handler functions:
 - arm_guard() 7-14

disarm_guard() 7-14
 dispatch_guard() 7-15
 pending_traps() 7-14
 which_guard() 7-15

trap mask functions:

mask_and() 7-7
 mask_clear() 7-7
 mask_fill() 7-7
 mask_in() 7-7
 mask_of() 7-7
 mask_or() 7-7

trap_of() 11-163
 TRAP_OF.C 11-163
 TRAPPER.C 7-20
 traps, defined 7-1
 2INT.C 11-153
 TXO operating system, defined 1-4
 TXO Workbench 1-4
 TYPE.C 11-135
 typographic conventions 1-6

U

UNIT_ID.C 11-121
 UNIX-V7 3-1
 unlink() 11-164
 UNLINK.C 11-164
 UNPK4.C 11-155
 USA font set E-3
 user traps 7-2
 USERTRAP.C 7-24

V

V command, modem off-hook 9-51
variable length record files 6-6
variable length records 6-24
VERSION.C 11-156
VFIRAW0.FDF 8-22
VFIRAW1.FDF 8-22
VFIRAW2.FDF 8-22
VISA 1st generation protocol 9-4, 9-6
VISA protocol 9-6
VLR files, compressed
 see CVLR files
VLR files:
 using 6-24
VLR.C 11-26
VLRFILE.C 6-25
volatile memory 3-8
volatile memory management 5-2
volatile memory, defined 5-2

W

W command, modem off-hook 9-52
WAIT.C 11-157
wherecur() 11-165
WHERECUR.C 11-165
wherewin() 11-166
wherewincur() 11-168
which_guard() 7-15, 11-169 – 11-170
which_guard(), using) 7-16
WHICHGRD.C 11-170
WHWINCUR.C 11-168
window() 11-171 – 11-172
WINDOW.C 11-167
windows:

pixel-type display 8-19
 pixel-type display, using 8-44
 segment-type display 8-14

Workbench

see *TXO Workbench*

write()

modem 9-30
 PIN Pad 9-22
 serial port 9-13
 write() function 11-173 – 11-174
 write():

bar code reader 8-61
 card reader 8-56
 clock 8-66
 display 8-12
 files 6-14
 keyboard 8-7
 LAN 10-22
 write_cmd() 11-175 – 11-176
 modem 9-32
 write_vlr() 6-14, 11-177 – 11-178
 write_vlr() 6-14, 11-179 – 11-180

X

X modem command 9-48

Z

ZAPD 1-4
 ZONTALK 2-2 – 2-3
 ZONTALK download 2-3
 ZONTALK download, LAN 10-39
 ZONTALK, CONFIG.SYS settings 4-10
 ZONTALK, downloading keyed files 6-21
 ZONTALK.C 11-159





Three Lagoon Drive
Redwood City, CA 94065-1561
TEL: 415-591-6500
FAX: 415-598-5504