Index

Table E.5: VT2XX Keys

VT2XX Key	Assigned to for downlodable keys	VT2XX Keycode	Sends (default on a VT2XX)
Find			ESC [1~
Insert here			ESC [2~
Remove			ESC [3~
Select			ESC [4~
Prev Screen			ESC [5~
Next Screen			ESC [6~
F6 (F6)	F1	17	ESC [17~
F7 (F7)	F2	18	ESC [18~
F8 (F8)	F3	19	ESC [19~
F9 (F9)	F4	20	ESC [20~
F10 (F10)	F5	21	ESC [21~
F11 (F11)	F6	23	ESC [23~
F12 (F12)	F7	24	ESC [24~
F13 (F13)	F8	25	ESC [25~
F14 (F14)	F9	26	ESC [26~
Help (F15)	F10	28	ESC [28~
Do (F16)	Shift F1	29	ESC [29~
F17 (F17)	Shift F2	31	ESC [31~
F18 (F18)	Shift F3	32	ESC [32~
F19 (F19)	Shift F4	33	ESC [33~
F20 (F20)	Shift F5	34	ESC [34~

Table E.4: VT100 Keypad (VT52 mode)

VT100 Key	UniTerm key	Numeric mode	Application mode
PF1	Keypad (	ESC P	ESC P
PF2	Keypad)	ESC Q	ESC Q
PF3	Keypad /	ESC R	ESC R
PF4	Keypad *	ESC S	ESC S
Keypad 7	Keypad 7	7	ESC ?w
Keypad 8	Keypad 8	8	ESC ?x
Keypad 9	Keypad 9	9	ESC ?y
Keypad 4	Keypad 4	4	ESC ?t
Keypad 5	Keypad 5	5	ESC ?u
Keypad 6	Keypad 6	6	ESC ?v
Keypad 1	Keypad 1	1	ESC ?q
Keypad 2	Keypad 2	2	ESC ?r
Keypad 3	Keypad 3	3	ESC ?s
Keypad 0	Keypad 0	0	ESC ?p
Keypad -	Keypad -	-	ESC ?m
Keypad ,	Keypad +	,	ESC ?l
Keypad .	Keypad .	.	ESC ?n
Keypad Enter	Keypad Enter	CR / CRLF	ESC ?M

Table E.3: VT100 Keypad

VT100 Key	UniTerm key	Numeric mode	Application mode
PF1	Keypad (	ESC OP	ESC OP
PF2	Keypad)	ESC OQ	ESC OQ
PF3	Keypad /	ESC OR	ESC OR
PF4	Keypad *	ESC OS	ESC OS
Keypad 7	Keypad 7	7	ESC Ow
Keypad 8	Keypad 8	8	ESC Ox
Keypad 9	Keypad 9	9	ESC Oy
Keypad 4	Keypad 4	4	ESC Ot
Keypad 5	Keypad 5	5	ESC Ou
Keypad 6	Keypad 6	6	ESC Ov
Keypad 1	Keypad 1	1	ESC Oq
Keypad 2	Keypad 2	2	ESC Or
Keypad 3	Keypad 3	3	ESC Os
Keypad 0	Keypad 0	0	ESC Op
Keypad -	Keypad -	-	ESC Om
Keypad ,	Keypad +	,	ESC Ol
Keypad .	Keypad .	.	ESC On
Keypad Enter	Keypad Enter	CR / CRLF	ESC OM

Appendix E

Key assignments and generated codes

Table E.1: VT100 Cursor Keys

VT100 Key	UniTerm key	ANSI normal	ANSI application
Cursor Up	Cursor Up	ESC [A	ESC OA
Cursor Down	Cursor Down	ESC [B	ESC OB
Cursor Right	Cursor Right	ESC [C	ESC OC
Cursor Left	Cursor Left	ESC [D	ESC OD

Table E.2: VT52 Cursor Keys

VT100 Key	UniTerm key	VT52 normal	VT52 application
Cursor Up	Cursor Up	ESC A	ESC A
Cursor Down	Cursor Down	ESC B	ESC B
Cursor Right	Cursor Right	ESC C	ESC C
Cursor Left	Cursor Left	ESC D	ESC D

- Leave UniTerm and start KeyEdit
- Load your old setup file
- Save on top of `UNITERM.SET`

Appendix D

The KeyEdit Program

If you have got a complete release of UniTerm, you should have a copy of `KEYEDIT.PRG` (Please check that it is for version 1.6f or higher!). This program enables you to edit the keyboard table stored in an UniTerm setup file, additionally you can create an executable keyboard table loader or just output the table itself.

D.1 Editing a UniTerm Setup File

Start `KEYEDIT.PRG` and press ⟨Return⟩ when you see the opening dialog box. It will take a few seconds to create the display. You can only edit a existing setup file with predictable results, so select [Load UniTerm File] and select the file you want to edit. Select the key you want to change (select the appropriate part of the table (normal, shifted or capslocked)) and then the character you want to assign to the key. When you are finished with editing, select [Save UniTerm File].

Don't try to change the characters on the keypad, cursorkeys or functionkeys in the UniTerm setup file. These keys are hardwired via keycode to specific strings (like on a real VT100).

D.2 Updating your Setup File

To move your custom keyboard table from a old version of UniTerm to a new one, do the following:

- Rename the old setup file
- Start UniTerm and set all parameters that need changing
- Save the setup as `UNITERM.SET`

SF4 = 51
SF5 = 52
SF6 = 53
SF7 = 54
SF8 = 55
SF9 = 56
SF10 = 57
Utilities = 58
ToggleMeta = 59
Help = 60
InsertClip = 61
SaveClip = 62
ViewHistEOL = 63

AutoPrint =	7
Zoom =	8
132ColumnToggle =	9
ScrollLock =	10
49LineToggle =	11
SendAnswerBack =	12
ShortBreak =	13
DropDTR =	14
LongBreak =	15
SaveHistory =	16
ControlHistory =	17
Switch =	18
ControlCapture =	19
PlayBack =	20
SendFile =	21
DegasSave =	22
ViewHistory =	23
Hangup =	24
Dial1 =	25
Dial2 =	26
Dial3 =	27
Dial4 =	28
Dial5 =	29
Dial6 =	30
Dial7 =	31
Dial8 =	32
Dial9 =	33
Dial10 =	34
SetPath =	35
DelFile =	36
DiskSpace =	37
F1 =	38
F2 =	39
F3 =	40
F4 =	41
F5 =	42
F6 =	43
F7 =	44
F8 =	45
F9 =	46
F10 =	47
SF1 =	48
SF2 =	49
SF3 =	50

```
# $2 string to wait for (CONNECT)
#
:4
set(1,3)
:5
   echo('\r\nTry: ')
   echo("add(4,-@1))
   set(1,add(@1,-1))
   wait(10)
# send attn string
   echo('\r\nSending +++')
   send('+++')
# we might get a OK here...
   get('OK',2)
   echo('\r\nSending ')
   echo($1)
   send($1)
   send('\r')
if(and(!get($2,@2),@1)) jump(5)
return()
############################################################
#                                                          #
# Home cursor and clear screen                             #
#                                                          #
############################################################
:20
echo('\033[f\033[2J')
return()
############################################################
```

A very useful aspect of the macro processor, is that you can assign a string like `%R('TEMPUS.PRG',' ')` to a function key and run your favourite editor just by pressing one key[2].

C.8 UniTerm Internal Function Numbers

ResetTek =	1
VDIOutput =	2
PrintTextScreen =	3
TekMode =	4
TextMode =	5
Reset =	6

[2]Remember that enough system memory has to be reserved for programs to run in.

```
# (in real life you would use the dial() function)
#
:2
# Get number from user
if(!input('Enter number (2512002)')) exit(0)
# if the string is empty use the default
if(compare($T,'')) concat('2512002','')
# Hayes want a ATD
concat('ATD',$T)
# dial.............
copy(1,$T)
copy(2,'CONNECT')
set(2,30)
call(4)
if(!@1) echo('\r\nFailed\r\n')exit(1)
#
# Login
#
# this is for our LocalNet 20 system
:6
send('\r')
if(!get('#',2)) break(100,FALSE) if(!get('#',2)) send('\001\001')
send('echo off\r')
if(!get('#',2)) echo('\r\nSomething is wrong!') hangup() exit(-1)
# call the system
send('call e780\r')
# should have a counter here, but you can always stop with ^C
if(!get('COMPLETED',3)) send('done\r') jump(6)
if(!get('ogin',3)) send('done\r') jump(6)
send('poole\r')
#
# should send my password here
#
# Finished!
exit(0)
############################################################
#                                                          #
#  Send a string to a Hayes compatible modem               #
#                                                          #
############################################################
#
# @1 number of retries
# @2 timeout
# $1 string to send (telephone number...)
```

C.6.3 Additional Statements

These statements can only be used in macro file mode, and will cause an error if used from a function key.

`if(`*int*`)`
execute the rest of the line if *int* is not equal 0

`jump(`*nr*`)`
goto label number *nr*

`exit(`*int*`)`
stop processing and return with value *int*

`call(`*nr*`)`
execute subroutine at label *nr* (subroutines can't be nested!)

`return()`
return from subroutine

C.7 Example

```
#####################################
# Test Macro for UniTerm V2.0c 002  #
# Copyright 1988 Simon Poole        #
#                                   #
#####################################
# turn history recording on
history(1)
call(20)
echo('\007\r\nSample UniTerm macro file')
echo('\r\n------------------------\007')
#
# Reset the modem
#
copy(1,'ATZ')
copy(2,'OK')
set(2,5)
call(4)
#
if(@1) echo('\r\nReset\r\n') jump(2)
#
echo('\r\nToo many retries\r\n') exit(-1)
#
# Dial the number and wait for CONNECT
```

C.4 String Constants

String constants are a maximum of 80 characters long and are enclosed in single quotes. Special ASCII values can be entered with the escape character \; every character after \ equals itself, except:

r	: is CR
n	: is LF
0xx	: \0xx is the octal ASCII value 0xx

C.5 Predefined String Variables

`$PATH`
contains the last file selector path, is initialized to the home directory of UniTerm.

`$FILENAME`
contains the last file selector filename, is initialized to ''.

`$CURRENT`
contains the current GEM DOS path.

`$TEMP`
tempory string for use in the macro processor.

`$VAR`
holds the address of UniTerm's parameter block[1] for passing to other programs.

C.6 Additional Features in Macro File Mode

C.6.1 Labels

Twenty local labels (per macro file) can be used: `:1` to `:20`. A label must be the first and only word on a line!

C.6.2 Comments

A line starting with # is ignored on input.

[1] Please consult the separate documentation on this subject.

`SET(`*nr,int*`)`
Set integer variable number *nr* (integer) to *int* (integer)
Returns: 0

`SUSPEND()`
Displays "Press any key..." on the statusline and waits for a key-press.
Returns: 0

`WAIT(`*time*`)`
Wait for *time* (integer) mSec×100.
Returns: 0

`UNICOMMAND(`*command*`)`
Execute one of UniTerm's internal commands, see list.
Returns: 0

`XMODEM(`*mode,file*`)`
Start XModem with:

mode (string):	`SEND` : send file
	`REC` : receive file
file (string):	file to send/receive

Returns: 0

`YMODEM(`*mode,filespec*`)`
Start YModem with:

mode (string):	`SEND` : send file(s)
	`REC` : receive file(s)
filespec (string):	file specification with wildcards

Returns: 0

C.3 Prefix Operators

`-`	negate integer value
`!`	logical not
`"`	return integer as string
`.`	return string as integer

Bug: `.` and `"` don't know about negative integers!

`LOADTEL(`*filename*`)`
Load a dialer setup file.
Returns: 0

`MESSAGE(`*msg*`)`
Displays *msg* (string) on the statusline. If *msg* is empty the normal statusline is restored.
Returns: 0

`MACRO(`*name*`)`
Execute the macro with filename *name* (string) from disk, default path is the current directory
Returns: the value of the exit command, or

-3 : syntax error (line in $TEMP)
-2 : not enough stack (nested more than one level)
-1 : macro buffer full (more than 4kB).

`OR(`*int1,int2*`)`
Logical or.
Returns: *int1* or *int2*

`PATH(`*path*`)`
Change default path to *path* (string).
Returns: 0

`POPUP(`*entry,command,name*`)`
Redefines an entry in the popup menu:

entry :	1..20
command :	see list
name :	string that will be displayed

Returns: 0

`RUN(`*name,commandline*`)`
Execute program *name* (string) with *commandline* commandline (string). If its a `.TTP` program you'll be asked for parameters.
Returns: Return code of program.

`REASSIGN(`*alt-key,command*`)`
Assigns integer value *command* to the alternate key value *alt-key*.
Returns: 0

`SEND(`*out-string*`)`
Send *out-string* (string).
Returns: 0

`ECHO(`*message*`)`
Echo *message* (string).
Returns: 0

`FILESELECTOR(`*path,filename,prompt*`)`
Show a fileselector with path *path* (string), filename *filename* (string) and a prompt of *prompt* (string). The new values for path and filename are in the variables `$PATH` and `$FILENAME`.
Returns: 1 if [OK] is selected, else 0.

`GET(`*in-string,time*`)`
Wait for *in-string* (string), with timeout *time* (integer) sec
Returns: 1 if sucsessful.

`HANGUP()`
Hangup the modem.
Returns: 0

`HISTORY(`*switch*`)`
Turn history recording on if *switch* is 1 (does not reset the buffer).
Returns: 0

`INLINE(`*mode*`)`
Reads a line from the keyboard (until ⟨Return⟩ is pressed or a maximum of 80 characters are read, ⟨Control⟩⟨C⟩ aborts), if *mode* (integer) is 1 (true) the characters are echoed. ⟨Delete⟩ and ⟨Backspace⟩ cause a destructive backspace. The line read is copied into `$TEMP`.
Returns: 0

`INPUT(`*prompt*`)`
Prompt for a line of input, result is in `$TEMP`, *prompt* is a string.
Returns: 1 if [OK] is selected.

`KERMIT(`*mode,8-bit-mode,filespec*`)`
Start Kermit filetransfer with the following parameters:

mode (string):	`SEND` : send file(s)
	`REC` : receive file(s)
	`GET` : receive file(s) from server
8-bit-mode (integer):	`0` : 7 bit ASCII text
	`1` : 8 bit binary
filespec (string):	file specification with wildcards

Returns: 0

`LOADSETUP(`*name*`)`
Load the setup file with name *name* (string).
Returns: 0

Function
any of the built-in functions can be used as an argument

C.2 Functions

All commands only have to be specified to the point they are unique (for most commands this means one character). Case is not significant.

`ADD(`*int1,int2*`)`
Arithmetic sum.
Returns: *int1* + *int2*

`AND(`*int1,int2*`)`
Logical and.
Returns: *int1* and *int2*

`ASSERT()`
Assert DTR.
Returns: 0

`BREAK(`*len,drop-dtr*`)`
Send Break for *len* (integer) mSec, drop DTR according to the *drop-dtr* (integer) value.
Returns: 0

`CONCAT(`*str1,str2*`)`
Concatenate *str1* (string) and *str2* (string) and put the result in the variable `$TEMP`.
Returns: 0

`COPY(`*nr,str*`)`
Copy *str* (string) to string variable number *nr* (integer).
Returns: 0

`COMPARE(`*str1,str2*`)`
Compare *str1* (string) to *str2* (string).
Returns: 1 if *str1* equals *str2* else 0.

`DROP()`
Drop DTR.
Returns: 0

`DIAL(`*num*`)`
Dial number *num* (1 to 10).
Returns: 1 if sucessful, 0 otherwise.

Appendix C

Simple Macro Processor

The macro commands can be in the strings assigned to the function keys or in a file, the command initiator is **%** for function keys (this is not needed in a macro file).

A command line is a maximum of 80 characters long. The macro-processor works in two modi:

Function-key mode
: all characters between commands are sent to the serial port

Macro file mode
: everything between commands except white-space is a syntax error, additional commands are available in this mode.

C.1 Arguments

Arguments are enclosed in parentheses and separated by commas, they can be of the following types:

String constant
: character string enclosed in single quotes, max. 80 characters.

String variable
: 10 user setable string variables are available: $1 to $10, additionaly 5 predifined strings can be used.

Integer constant
: positive integer in the range 0..32767.

Integer variable
: 10 user setable integer variables are available: @1 to @10, range: -32768 to +32767.

B.2 8-bit Control Codes

Dec	Oct	Hex	Keys ⟨Meta⟩ + ⟨Control⟩ +	7-bit aequiv.	ASCII name
128	200	80	⟨@⟩	ESC @	Unused
129	201	81	⟨A⟩	ESC A	Unused
130	202	82	⟨B⟩	ESC B	Unused
131	203	83	⟨C⟩	ESC C	Unused
132	204	84	⟨D⟩	ESC D	IND Index
133	205	85	⟨E⟩	ESC E	NEL New line
134	206	86	⟨F⟩	ESC F	SSA
135	207	87	⟨G⟩	ESC G	ESA
136	210	88	⟨H⟩	ESC H	HTS Horizontal tab set
137	211	89	⟨I⟩	ESC I	HTJ
138	212	8A	⟨J⟩	ESC J	VTS
139	213	8B	⟨K⟩	ESC K	PLD
140	214	8C	⟨L⟩	ESC L	PLU
141	215	8D	⟨M⟩	ESC M	RI Reverse index
142	216	8E	⟨N⟩	ESC N	SS2 Single shift 2
143	217	8F	⟨O⟩	ESC O	SS3 Single shift 3
144	220	90	⟨P⟩	ESC P	DCS Dev. ctrl string
145	221	91	⟨Q⟩	ESC Q	PU1
146	222	92	⟨R⟩	ESC R	PU2
147	223	93	⟨S⟩	ESC S	STS
148	224	94	⟨T⟩	ESC T	CCH
149	225	95	⟨U⟩	ESC U	MW
150	226	96	⟨V⟩	ESC V	SPA
151	227	97	⟨W⟩	ESC W	EPA
152	230	98	⟨X⟩	ESC X	Unused
153	231	99	⟨Y⟩	ESC Y	Unused
154	232	9A	⟨Z⟩	ESC Z	Unused
155	233	9B	⟨[⟩	ESC [	CSI Command seq.intro.
156	234	9C	⟨\⟩	ESC \	ST String terminator
157	235	9D	⟨]⟩	ESC]	OSC
158	236	9E	⟨^⟩	ESC ^	PM
159	237	9F	⟨_⟩	ESC _	APC

Due to the way the german bios is written (and probably most of the none US versions) not all of these codes can be produced with the keyboard.

B.1 7-bit Control Codes

Dec	Oct	Hex	Keys ⟨Control⟩ +	Atari character	ASCII name
0	000	00	⟨@⟩	None	NUL Null
1	001	01	⟨A⟩	Up Arrow	SOH Start of header
2	002	02	⟨B⟩	Down "	STX Start text
3	003	03	⟨C⟩	Right "	ETX End text
4	004	04	⟨D⟩	Left "	EOT End of trans.
5	005	05	⟨E⟩	Close Box	ENQ Enquiry
6	006	06	⟨F⟩	Move Box	ACK Acknowledge
7	007	07	⟨G⟩	Full Box	BEL Bell
8	010	08	⟨H⟩	Check	BS Backspace
9	011	09	⟨I⟩	Clock	TAB Horizontal tab
10	012	0A	⟨J⟩	Bell	LF Linefeed
11	013	0B	⟨K⟩	Note	VT Vertical tab
12	014	0C	⟨L⟩	FF	FF Formfeed
13	015	0D	⟨M⟩	CR	CR Carrige return
14	016	0E	⟨N⟩	Left Atari	SO Shift out
15	017	0F	⟨O⟩	Right "	SI Shift in
16	020	10	⟨P⟩	Led 0	DLE Data link escape
17	021	11	⟨Q⟩	Led 1	DC1 X-on
18	022	12	⟨R⟩	Led 2	DC2
19	023	13	⟨S⟩	Led 3	DC3 X-off
20	024	14	⟨T⟩	Led 4	DC4
21	025	15	⟨U⟩	Led 5	NAK Neg. acknowledge
22	026	16	⟨V⟩	Led 6	SYN Synchronus idle
23	027	17	⟨W⟩	Led 7	ETB End trans. blocks
24	030	18	⟨X⟩	Led 8	CAN Cancel
25	031	19	⟨Y⟩	Led 9	EM End of medium
26	032	1A	⟨Z⟩	?	SUB Substitute
27	033	1B	⟨[⟩	ES	ESC Escape
28	034	1C	⟨\⟩	Face p. 1	FS File sep.
29	035	1D	⟨]⟩	Face p. 2	GS Group sep.
30	036	1E	⟨^⟩	Face p. 3	RS Record sep.
31	037	1F	⟨_⟩	Face p. 4	US Unit sep.
32	040	20	⟨Space⟩		SP Space

Appendix B

ASCII Control Codes

	*NI Change alpha cursor colors
ESC TF colors	*NI Change color indices for dialog area
ESC TG plane colors	*NI Change color indices for plane
ESC TC first-color second-color third-color	
	*NI Change GIN cursor color
ESC SX number position	
	*NI Change GIN cursor position
ESC SUB	Enter 4010 GIN mode
ENQ	*NI Inquire 4105 status
US	*NI Enter 4105 alpha mode
ESC CAN	Enter echo suppression mode
FS	Enter marker mode
GS	Enter vector mode
ESC FF	Clear screen, enter alpha mode
ESC #!0	Report terminal mode
ESC ENQ	Report 4010 status
ESC %! mode	Change to a different terminal mode
ESC font	Change fonts
ESC style	Change 4014 line style
ESC ETB	4014 hardcopy

ESC MT text-color *NI Change text color

ESC NM mode *NI Prompt mode
ESC NR transmit receive
*NI Change baud rates
ESC NK time *NI Change break time
ESC NU char *NI Change echo suppression cancel character
ESC NE string *NI Change EOF string
ESC NT string *NI Change EOL string
ESC NC first-char second-char
*NI Change EOM characters
ESC NF mode *NI Change flow control mode
ESC NP string *NI Change prompt string
ESC NQ size *NI Change input buffer size
ESC NB stopbits *NI Change number of stop bits
ESC ND delay *NI Change transmit delay

ESC QI values *NI Map color to monochrome values for print
ESC QU density *NI Choose color hardcopy image density
ESC QD type *NI Choose printer type
ESC QL pages page-origin ff-mode
*NI Change dialog hardcopy attributes
ESC QE attributes *Ni Change hardcopy monochrome attributes
ESC QO orientation
*NI Change image orientation

ESC RU plane writing-mode bits-per-pixel
*NI Begin pixel operations
ESC RX dest.-plane d.-lower-left-corner
first-source-corner second-s.-corner
*NI Copy pixels
ESC RP number color
*NI Write pixels
ESC RR lower-left-corner upper-right-corner fill-color
*NI Fill rectangle
ESC RL array *NI ?
ESC RH position *NI Move to pixel position
ESC RS lower-left-corner upper-right-corner
*NI Change coordinates for pixel operations
ESC RW first-corner opposite-corner
*NI Change window on 4096*4096 coordinates

ESC TD first-color second-color

ESC KW mode *NI Enable key expansion
ESC KX number *NI Expand macro
ESC KH mode *NI Hardcopy
ESC KI mode Ignore deletes mode
ESC KF mode *NI Change line feed/carriage return mode
ESC KL mode *NI Lock keyboard
ESC KQ *NI Report errors
ESC KV *NI Reset
ESC KU *NI Save nonvolatile parameters
ESC KE mode *NI Change echo mode
ESC KZ char-delete line-delete literal
*NI Change edit characters
ESC KT threshold *NI Change error threshold
ESC KY char *NI Change key execute character
ESC KS mode *NI Change transparent mode
ESC KB positions *NI Change tab stops

ESC LL number *NI Change number of lines in dialog area
ESC LP start-point fill-boundary
*NI Start panel boundary
ESC LZ *NI Clear dialog area
ESC LG position *NI Draw to position
ESC LH position *NI Draw marker at position
ESC LE *NI Finish panel
ESC LT text *NI Graphic text
ESC LF position *NI Move to position
ESC LB number *NI Change number of lines for dialog buffer
ESC LI char-color char-background-color dialog-background-color
*NI Change dialog area color
ESC LV mode *NI Change dialog area visibility
ESC LM writing-mode
*NI Change dialog area writing mode

ESC MP number *NI Choose fill pattern
ESC MG writing-mode
*NI Change graphics area writing mode
ESC MN direction *NI Change graphtext direction
ESC MR mantissa power-of-two
*NI Change graphtext rotation
ESC MC width height spacing
*NI Change graphtext size
ESC ML color *NI Change line color
ESC MV style Change line style
ESC MM type Change marker type

A.4.4 Point Plot/Special Point Plot Mode

address	Plot point
intensity address	*NI Special plot point
CR	Alpha mode
ESC FF	Alpha mode

A.4.5 Incremental Point Plot Mode

SP	Pen up
P	Pen down
D	North
E	North east
A	East
I	South east
H	South
J	South west
B	West
F	North west
CAN	Alpha mode
ESC SUB	GIN mode

A.5 4105 Commands

In the following list capital letters are part of the command sequences and lower case identifiers denote Tektronix encoded parameters.

ESC IQ code	*NI Report terminal settings
ESC IJ normal shifted	*NI Change GIN cursor speed
ESC JC	*NI Copy
ESC KC	*NI Cancel
ESC KR mode	*NI Change carriage return/line feed mode
ESC KD number contents	*NI Define macro
ESC KO number contents	*NI Define nonvolatile macro
ESC KA mode	*NI Enable dialog area

Mode Changing

GS	Vector mode
ESC SUB	GIN mode
FS	Point plot mode
ESC FS	*NI Special point plot mode
RS	Incremental mode
CAN	Return to text terminal

A.4.2 Other Functions and Extended Escape Codes

ESC ETB	Hardcopy
ESC ENQ	Transmit status
ESC STX	Enable block fill/erase
ESC ETX	Disable block fill/erase
ESC \R	Enable rectangle draw
ESC \r	Disable rectangle draw
ESC x	Enable selective erase
ESC /0d	Dots on
ESC /1d	Dots off
ESC /2d	Dots complemented

A.4.3 Vector Mode

GS	Next vector is dark
address	Draw vector, next vector is visible
ESC ‘	Solid vector
ESC a	Dotted
ESC b	Dot-dashed
ESC c	Short-dashed
ESC d	Long-dashed
ESC p	Solid vector, XOR
ESC q	Dotted, XOR
ESC r	Dot-dashed, XOR
ESC s	Short-dashed, XOR
ESC t	Long-dashed, XOR
ESC SUB	GIN mode
US	Alpha mode, don’t move
CR	Alpha mode
ESC FF	Alpha mode, clear screen

Primary selector

(	G0
)	G1
*	G2
+	G3

Final selector

A	British
4	*RI Dutch
5 or C	Finnish
R	*RI French
9 or Q	French Canadian
K	German
Y	Italian
' or E or 6	Norwegian/Danish
%6	*NI Portugese
Z	Spanish
7 or H	Swedish
=	Swiss

A.4 4014 Mode

A.4.1 Alpha Mode

Cursor Movement

BS	Cursor left
HT	Cursor right
LF	Cursor down
VT	Cursor up
CR	Cursor to left margin

Character Set Sizes

ESC 8	Normal 35x76 (lines x columns)
ESC 9	Small 38x81
ESC :	Smaller 50x120
ESC ;	Smallest 58x133
ESC 6	*NI Enter Italics ?
ESC 7	*NI Exit Italics ?

ESA	Ignored
HTS	Horizontal tab set
HTJ	Ignored
VTS	Ignored
PLD	Ignored
PLU	Ignored
RI	Reverse index
SS2	Single shift G2 -⟩GL
SS3	Single shift G3 -⟩GL
DCS	Device control string introducer
PU1	Ignored
PU2	Ignored
STS	Ignored
CCH	Ignored
MW	Ignored
SPA	Ignored
EPA	Ignored
CSI	Control sequence introducer
ST	String terminator
OSC	Ignored
PM	Ignored
APC	Ignored

A.3 VT3XX Functions

Currently only the implemented control sequences are listed.

A.3.1 Set Mode

ESC ?67h ⟨Backspace⟩ sends BS and ⟨Delete⟩ sends DEL

A.3.2 Reset Mode

ESC ?67l ⟨Backspace⟩ sends DEL and ⟨Delete⟩ sends BS

A.3.3 National Replacement Character Sets

In this mode the ASCII characters:

```
# @ [ \ ] ^ _ ` { | } ~
```

are mapped into characters of the international character set. Only one NRC can be active at one time, the format to select one is:

ESC *primary final*

A.2.4 Reports

From host	
CSI ⟩0c	Secondary device attribute response
From terminal	
CSI ⟩1;17;0c	VT220, Software version 1.7
From host	
CSI ?25n	Are the function-keys locked?
From terminal	
CSI ?20n	Unlocked
CSI ?21n	*NI Locked

A.2.5 Misc

ESC [*Pn*X	Erase Pn characters
ESC [*Pn*P	Delete Pn characters
ESC [*Pn*@	Insert Pn blanks
CSI !p	Soft reset
CSI ?38h	Tektronix mode
CSI ?38l	VT200 mode
CSI ?25h	Cursor on
CSI ?25l	Cursor off

A.2.6 Downloadable Function Keys

DCS *Pc*;*Pl*|*Ky1*/*St1*;.. ST

Pc = 0 clear all keys (ignored)
Pc = 1 don't clear keys (ignored)
Pl = 0 lock keys (ignored)
Pl = 1 don't lock keys (ignored)
Ky1 Key number (decimal)
St1 String (hex)

A.2.7 Downloadable Character Sets

Not implemented! Sorry.

A.2.8 Control Codes

IND	Index
NEL	Next line
SSA	Ignored

B	US-ASCII
0	Special graphics

Logical character set selection

ESC ~	Lock shift G1 -)GR
ESC n	Lock shift G2 -)GL
ESC }	Lock shift G2 -)GR
ESC o	Lock shift G3 -)GL
ESC |	Lock shift G3 -)GR
ESC N	Single shift G2 -)GL
ESC O	Single shift G3 -)GL

A.2.2 Terminal Modes

CSI 61"p	VT100 mode
CSI 62"p	VT200 mode, 8-bit
CSI 62;0"p	" "
CSI 62;1"p	VT200 mode, 7-bit
CSI 62;2"p	VT200 mode, 8-bit
ESCSP F	*NI Send only C0 codes
ESCSP G	*NI Send C1 codes

A.2.3 Selective Erasing and Attributes

CSI 22m	Bold off
CSI 24m	Underline off
CSI 25m	Blink off
CSI 27m	Inverse off
CSI 0"q	Erase protection off
CSI 1"q	Non-erasable
CSI 2"q	Erasable
CSI ?K	Cursor to EOL
CSI ?0K	
CSI ?1K	SOL to Cursor
CSI ?2K	Whole line
CSI ?J	Cursor to EOP
CSI ?0J	
CSI ?1J	SOP to Cursor
CSI ?2J	Whole screen

ETB	Ignored
NAK	Ignored
DLE	Ignored
XON	In XOn/XOff flow control mode
XOFF	intercepted by ST bios, otherwise ignored

A.1.19 Nonstandard Functions

ESC [‘	Lock Keyboard
ESC [b	Unlock Keyboard
ESC [*Pn*I	Move Pn tabs right
GS	Enter 4014 Vector mode
ESC %!0	Enter Tektronix alpha mode (4105)
ESC #!0	Report terminal mode (4105)
%! SPSP 1	Report: I am a ANSI terminal (4105)
ESC [?39h	Set 49 line mode
ESC [?39l	Set 24 line mode
ESC [?40h	Set meta mode
ESC [?40l	Reset meta mode
ESC [*c	Inquire UniTerm version and mode

Response:
ESC [**major*;*minor*;*release*;*max-row*;*max-col*c
ESC P*ustring* ESC \
Execute *string* with UniTerm's macro processor

A.2 VT2XX Functions

A.2.1 Character Sets

The format is: ESC *primary final*

Primary selector

(	G0
)	G1
*	G2
+	G3

Final selector

⟨	DEC International

A.1.16 Editing Functions

ESC [*Pn*P	Delete character
ESC [*Pn*L	Insert Line
ESC [*Pn*M	Delete Line

A.1.17 Print Commands

ESC [?5i	Enter auto print
ESC [?4i	Exit auto print
ESC [5i	Enter printer controller
ESC [4i	Exit printer controller
ESC [i	Print screen
ESC [0i	
ESC [?1i	Print cursor line

A.1.18 Other Control Characters

NUL	Ignored
SOH	Ignored
ETX	Ignored
EOT	Ignored
ENQ	Transmit answerback message
BEL	Bell
BS	Backspace
HT	Horizontal tab
LF	Linefeed or CR LF
VT	same as LF
FF	same as LF
CR	Carriage Return
SO	Shift to G1 character set
SI	Shift to G0 character set
DC1	Ignored (Alternate XOn)
DC3	Ignored (Alternate XOff)
CAN	Cancel
SUB	Cancel
DEL	Ignored
US	Ignored
RS	Ignored
FS	Ignored
SYN	Ignored
EM	Ignored

A.1.12 Reset

ESC c	Reset to default values

A.1.13 Tests

ESC #8	Fill screen with E's
ESC [2;*Ps*y	* Invoke tests

A.1.14 Keyboard LED's

ESC [0q	All off
ESC [*Ps*q	LED Ps on

A.1.15 VT52 Mode

ESC	Enter ANSI Mode
ESC =	Enter alternate keypad mode
ESC ⟩	Exit alternate keypad mode
ESC F	Select special graphics character set
ESC G	Select US/UK character set
ESC A	Cursor up
ESC B	Cursor down
ESC C	Cursor right
ESC D	Cursor left
ESC H	Cursor home
ESC Y*PlPc*	Direct cursor address
ESC I	Reverse line feed
ESC K	Erase to end of line
ESC J	Erase to end of screen
ESC Z	What are you?
ESC /Z	I am a VT52 (Response)
ESC ^	Enter auto print mode
ESC -	Exit auto print mode
ESC W	Enter printer controller mode
ESC X	Exit printer controller mode
ESC]	Print screen
ESC V	Print cursor line

ESC [3g	Clear all tabs

A.1.9 Line Attributes

ESC #3	Double-height top half
ESC #4	Double-height bottom half
ESC #5	Single-width single-height
ESC #6	*RI Double-width single-height

A.1.10 Erasing

In Line

ESC [K	Cursor to end of line
ESC [0K	
ESC [1K	Beginning of line to cursor
ESC [2K	Entire line

In Screen

ESC [J	Cursor to end of screen
ESC [0J	
ESC [1J	Beginning of screen to cursor
ESC [2J	Entire screen

A.1.11 Requests/Reports

Requests from host		Reports to host	
ESC [5n	Status	ESC [0n	Terminal OK
		ESC [3n	* Terminal not OK
ESC [6n	Cursor pos.	ESC [*Pl*;*Pc*R	Cursor position
ESC [c	What are you?	ESC [?1;*Ps*c	VT100, Ps options
ESC [0c		ESC [?6;*Ps*c	VT102, Ps options
ESC Z		ESC [?62;*Ps*c	VT200, Ps options
ESC [?15n	Printer status	ESC [?10n	Printer ready
		ESC [?11n	Printer not ready
		ESC [?13n	No printer

ESC [0x	Send terminal parameter report after setup
ESC [1x	Send only on request

2	*NI Alternate special graphics set ROM

A.1.4 Shift into Character Sets

SO	Locked shift G1
SI	Locked shift G0

A.1.5 Character Attributes

ESC [m	No attributes
ESC [0m	No attributes
ESC [1m	Bold
ESC [4m	Underline
ESC [5m	Blink (Light)
ESC [7m	Reverse

A.1.6 Scrolling Region

ESC [*Pt*;*Pb*r	Set scrolling region

A.1.7 Cursor Movement Commands

ESC [*Pn*A	Cursor up
ESC [*Pn*B	Cursor down
ESC [*Pn*C	Cursor right
ESC [*Pn*D	Cursor left
ESC [*Pl*;*Pc*H	Cursor position
ESC [*Pl*;*Pc*f	
ESC D	Index
ESC M	Reverse Index
ESC E	Next line
ESC 7	Save cursor
ESC 8	Restore cursor

A.1.8 Tab Stops

ESC H	Horizontal tab set
ESC [g	Tab clear
ESC [0g	Tab clear

ESC [?7h	Auto wrap on
ESC [?8h	* Auto repeat on
ESC [?9h	* Interlace on
ESC [?18h	Print form feed on
ESC [?19h	Print extent full screen

A.1.2 Reset Mode

ESC [2l	Keyboard unlocked
ESC [4l	Replace mode
ESC [12l	Local echo on
ESC [20l	New line mode off
ESC [?1l	Cursor key cursor mode
ESC [?2l	VT52 mode
ESC [?3l	80 column mode
ESC [?4l	Jump scrolling
ESC [?5l	Screen normal
ESC [?6l	Origin mode absolute
ESC [?7l	Auto wrap off
ESC [?8l	* Auto repeat off
ESC [?9l	* Interlace off
ESC [?18l	Print form feed on
ESC [?19l	Print extent scrolling region

A.1.3 Select Character Sets

The format is: ESC *primary final*

Where *primary* selects one of the four logical character sets (G0 to G3) and *final* selects the actual character set to be mapped into the logical set.

Primary selector

(	G0
)	G1

Final selector

A	UK national
B	US-ASCII
0	Special graphics
1	*NI Alternate character set ROM

Appendix A

Control Codes and Escape Sequences

This appendix list control codes and escape sequences that complete implementations of VT102/VT2XX and Tektronix 4014/4105 terminals should interpret and the functions they should initiate. * marks functions that are redundant or not possible on an Atari ST computer, *NI marks other not implemented functions (due to my laziness?), *RI uncomplete implementation of a function, please read the implementation notes for details.

Consult the ASCII table for numeric values of the control codes, *Ps*, *Pn*, *Pc* and *Pl* denote decimal values (ESC [10;10f postion cursor at text coordinates (10,10)).

The following control codes and commands, the syntax and the command interfaces as a whole could possibly be patented or/and copyrighted, please consider this list as "for information only". Commercial use is strictly forbidden.

A.1 ANSI/VT2XX/VT102/VT100 Mode

A.1.1 Set Mode

ESC [2h	Keyboard locked
ESC [4h	Insert mode
ESC [12h	Local echo off
ESC [20h	New line mode on
ESC [?1h	Cursor key application mode
ESC [?3h	*RI 132 column mode
ESC [?4h	*RI Smooth scrolling
ESC [?5h	Screen reverse
ESC [?6h	Origin mode relative

6.5.1 Alpha Mode

Restrictions: 1 margin, all (well nearly all) character sizes are the same only the spacing is different.

6.5.2 Vector Mode

6.5.3 Zoom Mode

Restrictions: No zooming of graph text, slow.

6.6 Problems

Sometimes the serial port seems to blocked (this has happened to me with other ST terminal emulators as well), the reason for this still hasn't been discovered (probably due to a bug in the ST Bios), try resetting the terminal if this happens.

Don't use the [Set RS232 Port Parameters] dialog in an editor or the like, for some mysterious reason the serial port outputs a delete character if you actually change something and exit with [OK](probably due to a bug in the ST Bios).

Don't try to use the underscore character in the file-selector dialogs (this is a bug in GEM, fixed in the so called Blitter-TOS).

6.3.2 C0/C1 Modes

Not implemented, that means UniTerm always sends C0 codes (7 bit) (this only a restriction for the escape sequences sent by the cursor and keypad keys, you *can* send 8 bit codes from the keyboard), received C1 codes will be interpreted correctly.

6.3.3 International character set

The international character set is the default GR set (→the characters that are between ASCII 128 and 256). For technical reasons use of the GR set is slow (the set is changed for every character!).

6.3.4 Downloadable character set

Not implemented (very resolution dependent, useless on the ST).

6.3.5 Downloadable function keys

The lock/unlock/erase parameters are ignored. The mapping of the keys is described in appendix E, maximum string length per key is 80 characters (DON'T FORGET THAT % IS A SPECIAL CHARACTER FOR UNITERM!).

6.3.6 Regis

Not implemented (very resolution dependent, useless on the ST).

6.4 Tektronix Mode

6.5 General

The Tektronix mode is still a bit in a mess. This will change with the using the 4105 command set, this implies that you should only use the vector mode of the 4010 emulation as all other 4010 commands will probably be removed from the program.

Right now scaling is done with respect to a 4010 with 1024*780 points, this probably will change for the 4105 (512*360) emulation. In Tek coordinates the screen measures 4096*3120 points (4105: 4096*3072 points).

For redrawing and zooming purposes the incoming characters are stored in a circular buffer. This is quite a memory saving way to store them[1], but on the other hand this means they have to be decoded every time the vectors are drawn.

[1] one could naturally store the decoded vectors

The carrier detect signal and the break bit on the Mfp are polled once per main loop and if they are present a appropriate 'LED' is displayed.

UniTerm needs one VBL interrupt slot and also installs its own mouse-movement interrupt handler in GIN mode, additionly VDI mouse-movement and mouse-button handlers are installed. The mouse-button handler may cause problems with programs that use both buttons, since it maps both to the left button (this is a workaround AES's inability to wait for a left or right mouse-button event).

Please note, that unlike other available products UniTerm does NOT use any undocumented locations or functions of the operating system.

6.2 VT102/VT100 Mode

6.2.1 Smooth Scrolling

Only works upwards (this is hopefully the only direction anybody really needs), this will be fixed the day I get a blitter.

6.2.2 132 Column Mode

Only 128 columns wide, this is due to the 640 pixel resolution of the ST in horizontal direction.

6.2.3 Double Height/Width Characters

Restrictions: no double width in color.

6.2.4 Extended character set in 8 bit mode

Using the GR character set will work, but is probably very slow due to the fact that the font has to be set/reset for every character.

6.3 VT200 Mode

6.3.1 VT200 function keys

There is no default assignment of the VT200 function keys to ST keys, except for the downloadable function key strings. If you need the default VT200 keys (→appendix E), build yourself a setup file with the right settings.

Chapter 6

Implementation Notes

6.1 General

Most of UniTerm is written in CCD/OSS Pascal and uses standard GEM, GEMDos, Bios and XBios calls. The exceptions are:

- Scrolling, this is done with a assembler routine instead of a raster operations.
- Character output in the 128 column, DW and DH modes, is done with TXTBLT (which doesn't help very much speedwise), all other terminal mode output is done with fast custom assembler routines, which are at least an order of magnitude faster when using text attributes than the corresponding TXTBLT calls.
- Some miscellaneous routines, like CRC calculation, supervisor peeks and pokes.

Timing information for the main loop of UniTerm:

0.2 mS	RS232 state, keyboard and mouse state
If characters can be read from the serial port:	
0.18 mS	Cursor on/off (disabling the cursor saves this)
Innerloop (max. 20 iterations):	
0.1 mS	Character read
0.3 mS	Character output (VT100 mode 80 columns, no attributes set)

Turning history recording on will naturally make the loop slower, scrolling speed is 38 lines per second.

Chapter 5

Customizing UniTerm

UniTerm can be adjusted to suit your needs in various ways, most use the setup files to store the configuration data. The popup menu and the bindings of the ⟨Alternate⟩ keys are exceptions, mainly since there would have been no way of editing these settings without making UniTerm simply to large. The popup and keys can be set by executing UniTerm macro commands in the auto startup macro file.

5.1 Popup Menu

A line like

```
POPUP(1,15,'L. Break')
```

in your startup macro file will assign the command LongBreak to the top left-hand entry in the popup and will name the command 'L. Break', consult the chapter on the macro commands for a complete description.

5.2 Key Bindings

A line like

```
REASSIGN(39,15)
```

in your startup macro file will assign the command LongBreak to ⟨Alternate⟩ ⟨Space⟩ , to find out which keycode to use please consult other literature. Beware: in some cases the ⟨Alternate⟩ key modifies the value of the returned keycode. To override the default bindings, set the command number to zero, this causes the character generated by the BIOS to be used for output.

Timeout after
sets the maximum time Kermit waits for a character to be received from the serial port.

Maximum number of retries
the maximum number of retries before the transmission is aborted.

Number of padding characters
the number of padding characters sent before each packet.

Packet size
the maximum packet size you want to use (maximum size without large packets is 94 bytes, with 2048).

Quote character
the ASCII character used for control character quoting.

8 bit quote character
the ASCII character used for 8th-bit quoting.

Repeat prefix character
the character used for repeat prefixing.

Padding character
the character used for padding.

Start of packet
the character that marks the start of a Kermit packet.

IBM mode
wait for a XOn character before sending a packet (XOn/XOff flowcontrol naturally has to be turned off for this to work).

Error check type
two and single character checksum and CRC check.

Normally you shouldn't have to change any of these parameters, please consult the Kermit literature for more details.

UniTerm), if you have set parity to none and have selected 8 data bits, binary files will be transferred without 8th-bit prefixing; in all other cases prefixing will be used (be sure that the parity is the same on the complete connection to the remote computer). One note, since the Kermit "end of record" is the same as the TOS "end of line" no translation of CR 's or LF 's is done, this may lead to problems if you have a file on the ST which uses LF as "end of line" marker.

4.4.4 Server Commands

This is probably the simplest way to use Kermit, connect to the remote host and start the remote Kermit in server mode. A large subset of the possible Kermit server commands is implemented (not implemented: Journal, Who, Variable):

Put	send a file to the host.
Get	receive a file from the host.
Finish	terminate the remote Kermit.
Logout	terminate the remote Kermit and logout.
Dir	send a directory to the local Kermit and have it displayed on the screen (argument: file-spec).
Remote	send a command to the remote host (argument: commandline).
Type	send a file to the local Kermit and display it on the screen (argument: file).
CWD	change the current working directory of the remote Kermit (arguments: directory, password).
Usage	show disk usage (argument: area).
Program	start a program on the host computer (arguments: program-file, program-commands).
Erase	delete a file on the host computer (argument: file).
Copy	copy a file on the host computer (arguments: source, destination).
Rename	rename a file on the host computer (arguments: oldname, newname).
Login	login on a remote Kermit in server mode (arguments: user, password, account).
Help	get help from the remote host (argument: topic).
Status	get the current status of the server.

For more information consult the "Kermit Protocol Manual".

4.4.5 Setting the Kermit Parameters

The following parameters can be set:

Repeat count prefixing:	Yes
Alternate block checks:	Yes
Terminal emulation:	Yes (UniTerm)
Communication settings:	Yes (UniTerm)
Transmit BREAK:	Yes (UniTerm)
IBM mainframe communications:	Yes
Transaction logging:	No
Session logging:	No
Act as server:	No
Talk to server:	Yes
Advanced server functions:	No
Advanced commands for servers:	Yes
Local file management:	Yes
Handle file attributes:	Yes[2]
Command/init files:	Yes (UniTerm)
Command macros:	Yes (UniTerm)
Large packets:	Yes
Windows:	No

Please don't forget if you miss a feature, that UniTerm is a terminal emulator and not a "real" Kermit (whatever that maybe).

4.4.2 Simple File Transfer

To receive a file, type the correct command for the remote Kermit and type ⟨Alt⟩⟨T⟩ on the ST keyboard, a large dialog box should appear. Select [Receive] if you want to use a different name than the original filename use the file selector form to select a name (this will only effect the first file received in a mutiple file transfer). If you want to receive the files with the names supplied by the host, just press ⟨Return⟩ (the path entered is retained till the end of your UniTerm session). A new form will appear and will display the current file being received, the total number of packets, the current number of errors and timeouts and the last non-fatal error that occurred. The transfer can be aborted by typing ⟨Control⟩⟨C⟩ .

To send a single file or a group of files, setup the remote host for receiving, type ⟨Alt⟩⟨T⟩ and select [Send]. Enter the filename or wildcards (*, ?) in the file selector dialog and press ⟨Return⟩ .

4.4.3 Binary File Transfer

Set both sides (host and local computer) to binary mode (on most mainframes with `set file type binary` or `set file binary`, select the [Binary] button on

[2] "Handle" is the wrong word, all attribute types except the file size are ignored

4.3.2 Using YModem Batch

YModem is a version of the XModem protocol with CRC type block check and with a batch send/receive protocol added. The file length will be set to the value received in the header block, file date and time is set to 0 and the attribute to 000644 (octal) on sending and ignored on receiving. Filename collision handling and wildcard expansion are done on receiving/sending a group of files.

4.3.3 Setting the XModem Parameters

The following parameters can be set:

Timeout after
: sets the maximum time XModem waits for a character to be received from the serial port.

Maximum number of errors
: sets the maximum number of errors before the file transfer is aborted.

Accept ASCII NUL
: allows you to use XModem for none binary file transfers, when ASCII NUL is used as padding character.

Packet size
: chooses the packet size for XModem transfers, the 1kB size may be more efficient on noise free lines.

Error check type
: sets the default checksum type, if possible use the CRC check for the larger packet size.

4.4 Kermit

4.4.1 UniTerm Kermit Capabilities

UniTerm Kermit capabilities at a glance:

Local operation:	Yes
Remote operation:	No
Transfers text files:	Yes
Transfers binary files:	Yes
Wildcard send:	Yes
^X/^Y interruption:	No
Filename collision avoidance:	Yes
Can time out:	Yes
8th-bit prefixing:	Yes

End of file transfer
a string that is sent when the transfer finishes.

Send
determines if the output translation table is used for sending files.

Receive
determines if the input translation table is used for capturing files.

Delay Time
sets the time [ms] UniTerm waits after every character sent (this is implemented with the 200Hz system clock.

Method
if [Paced by Echo] is selected, UniTerm waits for every character sent to be echoed (except ASCII control codes), this doesn't time out!

Translate EOL to
selects if CR LF [1]is sent as CR LF , CR , LF or as SP CR . This function will send single CR 's and LF 's correctly!

Translation on input
allows you to change the translation table used during file capture

Translation on output
enables you to edit the translation table which is used for sending files, enter the decimal ASCII values of the characters or nothing if you want the character to be ignored.

4.3 XModem/YModem

4.3.1 Using XModem

To receive a file, start the remote XModem and type ⟨Alt⟩⟨T⟩ . To receive a file select [Receive] from the dialog box and enter the filename in the file selector form. If you have set the error check mode to [CRC] UniTerm will try to initiate a file transfer with CRC error check, if it gets no response in the maximum allowed number of errors, it will retry with the normal checksum. To send a file select [Send] and enter the name of the file in the file selector.

[1]The normal Atari ST end-of-line marker.

Chapter 4

File Transfer

4.1 General

To change the current file transfer mode use the [Transfer] menu. Changing the file transfer type here, changes the dialog box that is displayed when you select [File Transfer] from the [Settings] menu and what happens when you type ⟨Alt⟩⟨T⟩ (start file transfer).

4.2 ASCII File Transfer

4.2.1 Using ASCII File Transfer

There is nothing much to say about ASCII file transfer, just press ⟨Alt⟩⟨T⟩ and choose the file you want to send. Pressing ⟨Help⟩ gets you to the Help screen, so you can change the delay between characters to a different value during an upload (this is useful with VMS which normally has a lot of trouble with the first line sent). The transfer can be interrupted at any time by pressing ⟨A⟩ . Normally some experimentation is needed to find the shortest delay time for your system. You can use a character translation table to map incoming characters to Atari characters.

To receive files use the file capture function. A translation table is used in the same way as with sending files. When file capture is in effect the characters are displayed as they will be saved.

4.2.2 Setting the ASCII File Transfer Parameters

The following parameters can be set:

Start of file transfer
: a string that is sent before file transfer starts.

user are stored in a circular buffer. Pressing ⟨Insert⟩ starts the sle, the status line will clear and the current line will be displayed instead. The following functions are provided:

⟨→⟩ move the cursor one character to the right.

⟨←⟩ move the cursor one character to the left.

⟨↑⟩ display the previous line.

⟨↓⟩ display the next line.

⟨Delete⟩ delete the character under the cursor.

⟨Backspace⟩ delete the character to the left of the cursor.

⟨Insert⟩ terminate the sle without sending the current line.

⟨Return⟩ send the current line and leave the sle.

The sle is always in insert mode. **DON'T FORGET TO ERASE YOUR PASSWORDS!** If you want to disable the sle for this reason, disable it in the [Terminal] dialog box.

3.11 Single Character Commands

All functions in the "Help" menu of UniTerm can used with single character commands:

1	Set terminal parameters part 1
2	Set terminal parameters part 2
A	Set ASCII file transfer parameters
B	Set buffer sizes
C	Configure dialer
D	Delete file
E	Edit function-keys
F	Show free diskspace
G	Set graphic parameters
I	Show info dialog
K	Set Kermit parameters
L	Load setup
P	Set path
R	Run program
S	Save setup
T	Set tabs
Q	Quit "help" dialog
V	Set RS232 parameters
X,Y	Set X/YModem parameters

3.7 File Capture/Playback

The file capture routine uses a character translation table, which can be edited by selecting [ASCII] from the [Transfer] menu and then selecting [File Transfer] from the [Settings] menu.

While file capture is in progress all screen output is translated too, so you will get a direct impression of what you are saving.

The Playback function allows you to simulate input from the serial port, pressing ⟨A⟩ allows you to abort, any other key will cause UniTerm to wait for another keypress.

3.8 Editing Function Keys

To edit the string assigned to a function key, press the ⟨Help⟩ key and select [Edit Function Keys] from the [Other] menu. You can edit the strings now, the cursor keys will move you to the string you want to edit, ⟨Esc⟩ will clear the string, ⟨Backspace⟩ will delete to the left, ⟨Delete⟩ the current character. Control codes can be entered by pressing ⟨Control⟩ and the appropriate key (see appendix B).

3.9 The Statusline

The 25th line on the screen is used as a statusline in text terminal mode, it should look like this (all possible information shown):

UniTerm	V2.0c	Online	Meta	Caps	HPDBLCR	1234
Program name	Version	Mode	Meta Ind.	Caps Ind.	Status	LED

The "Status LED's" are (from left to right):

	On	Off
History	H	_
Autoprint	P	_
DCD	D	_
Break	B	_
Keyboard locked	L	_
File Capture	C	_
Insert Mode	I	R

3.10 The Single-line Editor

To make working on systems that don't have a single-line editor (short sle) easier, a simple sle is implemented in UniTerm. The last 20 lines typed by the

Aspect
this option controls the scaling of pictures.

Deletes
allows you to set processing of DEL in the 4010 module.

3.4 Tabs

This small dialog allows you to set the tab positions. Generally it is not a good idea to change them from the default settings, since there is a lot of (buggy) software that relies on the settings being the same as the original VT100 factory settings.

3.5 Changing Buffer Sizes

Select [Buffers] from the [Settings] menu, the top half of the dialog form allows you to change four values:

- Transfer buffer size
- RS232 input buffer size
- System reserved size
- Clipboard size

the [+] and the [−] buttons autorepeat, double clicking increases (decreases) the displayed value by 1000.

In the bottom half the actual amount of memory allocated to each buffer is displayed, if these values are not the same as the ones set by you, for some reason UniTerm was not able to use your configuration. In this case 5kB are reserved for system use, 2kB for the clipboard and the rest of available memory is allocated to the transfer buffer (up to the amount set by you, minimum 1kB), the remaining memory is used by the history buffer (min. 1kB).

The RS232 buffer values will only change if you save the value in a setup file and restart UniTerm, all other changes take effect immediately (and reset the buffers!).

3.6 Setup Files

All user setable parameters are stored in the setup files (including the function key strings etc.). Loading a setup file from a different version of UniTerm may result in a version conflict error message, if this happens UniTerm uses its internal defaults.

Delete
changes the way ⟨Backspace⟩ and ⟨Delete⟩ work, if set to [Delete] ⟨Backspace⟩ will send BS and ⟨Delete⟩ will send DEL , set to [Backspace] the codes are exchanged.

3.2.2 Terminal 2

Answerback
is the string which is sent as answerback message.

Printer
determines if a printer is connected or not.

Print terminator
determines if a FF is sent after each print operation or not.

SLE
turns the built-in single-line editor on and off.

Auto executed macro
The contents of any file named here, are executed on startup by the UniTerm macro processor

NRC
turns national replacement character set mode on, and controls which character set is used (selecting ASCII turns it off). [4]

3.3 Graphics Terminal Parameters

Select [Graphics] from the [Settings] menu to change the parameters of the graphic terminal module:

Tektronix mode
enables or disables automatic switching to the Tektronix mode after a GS character, useful on noisy telephone lines (this option also inhibits the allocation of a 32kB buffer for the Tektronix screen).

GIN Termination String
sets the string sent after a GIN report.

Status Termination String
sets the string sent after a status report.

[3] All control codes are displayed on the screen, instead of causing a screen operation
[4] Use this mode only if you are using UniTerm in a 7-bit enviroment, normally you should use the 8-bit character set for this.

[Echo] UniTerm echos user input on the screen.

[Local] User input is only sent to the screen.

3.2 Terminal Parameters

To change these settings, press ⟨Help⟩ and select [Terminal 1] or [Terminal 2] from the [Settings] menu, the following parameters can be changed:

3.2.1 Terminal 1

Terminal
selects the terminal mode and change the terminal identification. [4014] selects 4014 alpha mode, [DCM] selects Display Control Mode[3]

Keypad
chooses the mode of the keypad, in application mode escape sequences are sent by all keys, in numeric mode only the top 4 keys send escape codes (they are the keys F1 to F4 on a real VT100).

Cursormode
selects the two different sets of escape sequences that can be sent by the cursor keys.

Use
masks out the 8th bit of sent or received bytes for the terminal emulations, set to 8 bits if you want to use the 8 bit VT2XX mode (this only effects ASCII file transfer and not the other file transfer modes).

Scroll
selects between slow and quick scroll.

Newline Mode
sets the VT100/102/200 newline mode (what is sent when you press ⟨Return⟩).

Wrap
switches the automatic wraparound at the end of a line on and off.

Cursor
selects blinking or non blinking, underline or block cursor.

Background
sets the background (and naturally the foreground) color

[2]to stop people always asking what the "Full" string in the statusline means; "Online" is displayed instead

Chapter 3

More About UniTerm

This chapter contains a short description of all user-setable parameters, except those related to the various file transfer protocols. Please read the chapter 4 for more information.

3.1 RS232 Port Parameters

To change these settings, press ⟨Help⟩ and select [RS232] from the [Settings] menu. Following parameters can be changed:

Baud
selects the rate for the serial port.

Flowcontrol
selects the flowcontrol mode, these functions are implemented by the BIOS.

Parity
selects the parity mode, these modes are implemented by the BIOS[1].

Databits
allows you to select the number of data bits, implemented by the BIOS.

Stopbits
allows you to select the number of stop bits, implemented by the BIOS.

Mode
allows you to put the terminal in one of following modes:

[Full] Data typed on the terminal is echoed by the host computer[2].

[1] Basic Input Output System

setup file with KeyEdit to get all ASCII codes[11].

[11] typically { ,} ,[and] are missing

2.14 Meta Mode

In Meta mode ⟨Alternate⟩ is the so called Meta key; pressing ⟨Alternate⟩ plus a second key will produce the ASCII code of that letter plus 128 (the Meta key sets the eighth bit of the character). To enter Meta mode press ⟨Alt⟩⟨CapsLock⟩ (this will toggle an indicator on the statusline).

Some of the more important characters of the international character set that can be generated are:

⟨Meta⟩ +	Character	⟨Meta⟩ +	Character
⟨@⟩	À	⟨‘⟩	à
⟨A⟩	Á	⟨a⟩	á
⟨B⟩	Â	⟨b⟩	â
⟨C⟩	Ã	⟨c⟩	ã
⟨D⟩	Ä	⟨d⟩	ä
⟨E⟩	Å	⟨e⟩	å
⟨F⟩	Æ	⟨f⟩	æ
⟨G⟩	Ç	⟨g⟩	ç
⟨H⟩	È	⟨h⟩	è
⟨I⟩	É	⟨i⟩	é
⟨J⟩	Ê	⟨j⟩	ê
⟨K⟩	Ë	⟨k⟩	ë
⟨L⟩	Ì	⟨l⟩	ì
⟨M⟩	Í	⟨m⟩	í
⟨N⟩	Î	⟨n⟩	î
⟨O⟩	Ï	⟨o⟩	ï
⟨Q⟩	Ñ	⟨q⟩	ñ
⟨R⟩	Ò	⟨r⟩	ò
⟨S⟩	Ó	⟨s⟩	ó
⟨T⟩	Ô	⟨t⟩	ô
⟨U⟩	Õ	⟨u⟩	õ
⟨V⟩	Ö	⟨v⟩	ö
⟨W⟩	Œ	⟨w⟩	œ
⟨X⟩	Ø	⟨x⟩	ø
⟨Y⟩	Ù	⟨y⟩	ù
⟨Z⟩	Ú	⟨z⟩	ú
⟨[⟩	Û	⟨{⟩	û
⟨\⟩	Ü	⟨|⟩	ü
⟨]⟩	Ÿ	⟨}⟩	ÿ
⟨-⟩	ß		

If you have a non-US keyboard, you will probably have to edit your UniTerm

2.11 Popup Menu

Besides being bound to keys some functions of UniTerm are accessible via a popup menu[8]. Click the right mouse button (while the mouse cursor is visible) and the popup will appear at the current cursor position. To select one of the items just click the left button, to get rid of the menu click outside of the popup area. The default configuration assigns the ten telephone numbers of the dialer to the fields on the right-hand side.

2.12 Viewing the History Buffer

You can view the contents of the history buffer with the key combination ⟨Alt⟩⟨V⟩ [9]. Besides the normal 'clip' functions with the mouse, six keys have a special function:

⟨Undo⟩	Exit.
⟨Insert⟩	Bottom of buffer.
⟨ClrHome⟩	Top of buffer.
⟨↑⟩	Up one line.
⟨↓⟩	Down one line.
⟨→⟩	Up one page.
⟨←⟩	Down one page.

2.13 Dialer

The setup file `UNITERM.TEL` is loaded at startup, if it isn't found the values for the dialer are reset. The keys ⟨Alt⟩⟨1⟩ to ⟨Alt⟩⟨0⟩ dial numbers 1 to 10, ⟨Alt⟩⟨H⟩ sends the hangup string.

A "+" as first character of the number is a placeholder for the access code. Dialling can be aborted by pressing ⟨Control⟩⟨C⟩ , the timeout is 40 seconds.

The suffix, prefix and hangup strings are passed to the macro interpreter in "function-key"[10] mode.

If the number and macro field is emtpy, the dialer returns straight away, if the number field is empty and a macro file is specified, the macro file is executed.

A tip for people that want the macro to handle waiting for the "CONNECT" or whatever it may be message, just leave out the "Connect msg" entry. The dialer will then immediatly start executing the macro file after sending the number string.

[8] The default configuration can be changed with the macro command REASSIGN

[9] You must reserve at least 25kB of system memory for this to work, the memory will only be used as long as the view mode is active

[10] This means characters between the macro commands are sent aswell

⟨CapsLock⟩ Toggle Meta mode.

2.9 The Clipboard

Pressing the left mouse button while the I-type mouse cursor is visible[5] will produce a "rubber box", after you have let go of the mouse button the selected text will be inverted and a small popup menu will appear:

- clicking outside the popup will cancel the operation,
- selecting the [Cut] item with the left mouse button will store the text in a buffer (the "clipboard"). Doing the same with the right button will append a CR after each line,
- [Add] appends to the text already in the clipboard (with the same difference between left and right mouse button),
- [Send] sends the text directly to the host computer[6], without using the clipboard (same usage of left and right button).

Two commands supplement this feature, `SaveClip` and `Insert` which are available via the main popup menu:

- `SaveClip` allows you to save the contents of the clipboard to a file, CR 's are mapped to CR LF .
- `Insert` sends the contents of the clipboard to the host computer.

Additional operations on the clipboard are possible with seperate programs, the clipboard can be accessed via the UniTerm parameter block, example programs and documentation should be available with this manual.

2.10 Mouse Cursor Control

In practically all situations where you can use the cursor keys, you can use the mouse to position the cursor too; while the mouse cursor is visible (the I-type text cursor) move it to the new position and double click the left mouse button; the cursor should now move to the new position[7]. Some editors do not allow you to move the cursor over tabs, this will cause the cursor to miss the intended end position in some cases.

[4]DEGAS is a trademark of Batteries Included Inc.

[5]If it is not visible, move the mouse a bit

[6]The delay between the cursor key codes can be set with the "Delay" parameter in the "ASCII File Transfer" dialog

[7]The delay between the cursor key codes can be set with the "Delay" parameter

⟨F2⟩ Toggle 24/49 lines mode (only on monochrome monitor).

⟨F3⟩ Write history buffer contents to a VDI-device (printer or meta-file).

⟨F4⟩ Prints the contents of the textbuffer (this is a very quick way of getting a copy of the screen contents, it is much faster than using the normal screen dump).

⟨F5⟩ Switches to the graphics screen and sets the terminal to Tektronix 4010 mode.

⟨F6⟩ Switches the screen and the terminal to VT102/VT100 mode.

⟨F7⟩ Resets the terminal, reads the default values from disk.

⟨F8⟩ Toggle autoprint.

⟨F9⟩ Enter zoommode.

⟨F10⟩ Toggle between 132(128) and 80 column mode.

⟨A⟩ Send the answerback string.

⟨B⟩ Send a short break (0.233 s) (doesn't drop DTR).

⟨C⟩ Start/stop file capture.

⟨H⟩ Hangup the telephone.

⟨L⟩ Send a long break (3.5 s) (drops DTR).

⟨P⟩ Screen dump to disk in DEGAS[4] format.

⟨R⟩ Playback a file with the terminal emulator.

⟨S⟩ Control history recording.

⟨T⟩ Start file transfer (starts file transfer with the protocol selected in the [Transfer] menu).

⟨V⟩ View the history buffer.

⟨X⟩ Save history buffer to disk.

⟨Z⟩ Hold Screen (⟨Y⟩ on the german keyboard).

⟨1–0⟩ Dial numbers 1 to 10.

⟨Help⟩ Atari screen dump.

⟨Insert⟩ Start the single-line editor.

2.5 Zoom Mode

If your history buffer is large enough, you can redraw a picture with different scaling factors. To select this mode press ⟨Alt⟩⟨F9⟩ , the screen will be cleared, the current contents of the buffer will be drawn on the screen and the normal arrow mouse cursor will appear.

To select the region of the picture you want to magnify, move the arrow to the upper left corner of the region, press the left mouse button and drag the mouse to the lower right corner of the region (a "rubber box" should follow the arrow) and release the button. The resulting picture will be scaled so that the larger side of box will fit on the screen (if the option True Aspect is selected). You can repeat this operation until a magnification factor of thirty is reached.

⟨Backspace⟩ will restore the previous scaling factor, the arrow keys will move the screen a third of the screen width/height in their direction, ⟨Return⟩ will reset to the original scaling and ⟨Undo⟩ will leave zoom mode. Remember that if your picture is built out a lot of vectors, the redrawing may take some time!

2.6 GIN Mode

The sequence ESC SUB will enter GIN mode from any of the Tektronix modes, it will not work from the VT102/VT100 mode since SUB cancels all ANSI type commands. A crosshair cursor will appear which can be moved with the mouse [3]. Pressing any key on the keyboard will cause a GIN report to be sent to the host and exits the GIN mode to Tektronix alpha mode (if for some reason the program switching UniTerm into GIN mode does not stop, you can exit completely by pressing ⟨Undo⟩).

2.7 Tektronix 4014 Alpha Mode

This mode is only included for compatibility with the GIN-mode and is not very useful. It is the same as 4010 alpha mode with one margin. This mode will probably be replaced in future versions with a 4105 compatible alphamode.

2.8 Using the Special Keys

Here is a list of the functions on the special keys (press ⟨Alt⟩ and the key listed here to invoke the function):

⟨F1⟩ Erases the graphics screen and resets the Tektronix emulation from any terminal mode.

[3] Moving the mouse to quick may cause the mouse handler to miss some interrupts, resulting in some rather odd behaviour of the cursor.

the setup, select [Save setup] from the [File] menu, a normal GEM file selector dialog will appear, if you want to use this setup as default, save it with the name `UNITERM.SET` and UniTerm will load it automatically on startup. To leave the help screen select [Quit Menu], press ⟨Q⟩ or the left mouse button once.

If you have adjusted the parameters correctly, you should now be able to communicate with your host computer. Some operating systems (VMS) try to identify the terminal automatically, depending on your setup UniTerm will identify itself as a DEC VT200, VT102, VT100 or as an VT100 emulating a VT52. A VT102/VT200 has more "advanced" editing functions (which results in less overhead in transmitting inserts etc.) so leave this option on VT102/VT200 if possible. If you have to enter the terminal type manually try VT102 first (changing from VT102 to VT100 or to VT200 only changes the report from UniTerm, not the actual commands UniTerm understands).

2.3 Exiting UniTerm

To stop UniTerm, press the ⟨Undo⟩ key, an alertbox will appear asking for confirmation[2].

2.4 Vector Graphics Mode

After receiving the control code GS (this can be turned off) or the 4105 command ESC %!0 (ASCII control characters are printed in this font to distinguish them from printable characters, a space is written SP) from the host, UniTerm switches to a separate graphics screen, if the host doesn't send ESC FF (the ASCII characters Escape and Formfeed) before starting a new picture, you will have to clear the screen manually with ⟨Alt⟩⟨F1⟩ . The incoming characters will be interpreted as Tektronix encoded vectors or commands (see appendix A) until UniTerm receives one of the following codes:

- CAN will return you to VT102 mode,
- ESC FF will clear the screen and enter Tektronix alpha mode.
- ESC %!1 will return to VT102 mode.

To return manually from Tektronix to VT200/VT102/VT100/VT52 mode press ⟨Alt⟩⟨F6⟩ or select the appropriate mode from the item Terminal in the menu Settings. To view your picture again press ⟨Alt⟩⟨F5⟩ . Be careful, selecting Tektronix 4010 mode manually will also reset the history buffer!

[2] to avoid the alertbox, press ⟨**Alt**⟩⟨**Undo**⟩

[Transfer] selects the file transfer protocol you want to use: A change here is reflected in a different dialog box appearing when you select the [File Transfer] item in the [Settings] menu and in the protocol used when you type ⟨Alt⟩⟨T⟩ .

2.2.4 Settings Menu

Desk	File	Transfer	Settings	Other
			RS232	
			Terminal 1	
			Terminal 2	
			File Transfer	
			Graphics	
			Tabs	
			Buffers	

[Settings] allows you to select from:

[RS232]	sets the parameters of the serial port.
[Terminal 1]	sets the value of some terminal (text mode) parameters.
[Terminal 2]	sets the rest of the terminal specific parameters
[Graphics]	sets the values for the graphics terminal module of UniTerm.
[File Transfer]	allows you to set the parameters for the current file transfer mode.
[Buffers]	set the sizes of the buffers UniTerm uses.
[Tabs]	set the tab positions (do not change without need, a lot of software depends on the "factory" settings)

2.2.5 Other Menu

Desk	File	Transfer	Settings	Other
				Dialer
				Edit Function Keys

[Other] has two items:

[Edit Function Keys]	allows you to assign a string and/or macro commands to a function key.
[Dialer]	setup the dialling sequences and telephone numbers for your modem.

Select the [RS232] item in the [Settings] menu and a new dialog will appear. Selecting the different values for the parameters is quite straightforward, just click on the buttons with the right values (we hope you know them, otherwise you will just have to experiment) and when you are finished select [OK]. To save

2.2.1 Desk Menu

Desk	File	Transfer	Settings	Other
About Uniterm...				

[Desk] is the well known menu where you can start desk accessories and with the [About UniTerm...] entry you can find out which version of UniTerm you are using.

2.2.2 File Menu

Desk	File	Transfer	Settings	Other
	Load Setup Save Setup			
	Load Numbers Save Numbers			
	Show Space Set Path Delete File			
	Run Program			
	Quit			

[File] allows you to select from:

[Load Setup]	load a previously saved setup from disk.
[Save Setup]	save the current settings of UniTerm to disk.
[Load Numbers]	load a setup file for the dialer
[Save Numbers]	save a setup file for the dialer
[Delete File]	delete a file.
[Set Path]	change the current GEMDOS drive and path.
[Show Space]	show total available and free space on the current drive.
[Run Program]	execute a program without leaving UniTerm.
[Quit Menu]	leave this screen and return to terminal mode.

2.2.3 Transfer Menu

Desk	File	Transfer	Settings	Other
		ASCII XModem YModem Kermit		

Chapter 2

Starting Using UniTerm

2.1 Requirements

To use UniTerm you need:

- a ATARI ST computer
- a monochrome or color monitor
- a RS232 standard cable
- a computer/modem to connect to
- a floppy disk containing `UNITERM.PRG` and `UNITERM.RSC`

2.2 Getting Started

After connecting your ST to your host computer, double click the UniTerm icon. If you are using UniTerm for the first time an alert box will appear, press ⟨Return⟩ [1] and ignore the error message (UniTerm didn't find the file `UNITERM.SET`, which we will create later on), the screen will clear to white (on a monochrome monitor) with a statusline on line 25. Press ⟨Help⟩ and the UniTerm help screen and a menu bar with following contents will appear:

Desk	File	Transfer	Settings	Other

[1] in this manual ⟨xx⟩ means the the key with xx on it, ⟨Alt⟩ is short for ⟨Alternate⟩

Chapter 1

Introduction

UniTerm is a program that emulates most of the functions of DEC's (Digital Equipment Corporation) VT102 and VT220 text terminals and of Tektronix's 4014 graphics terminal. Additionally UniTerm provides the XModem, YModem and Kermit file transfer protocols.

The program UniTerm is copyrighted, it can be copied, distributed and used free of cost, but may not be sold for more than the actual distribution costs. Please contact me, if you find bugs or have suggestions for revised versions of UniTerm, but read appendix A and the implementation notes first.

Some compromises have been made in the implementation of the VT100 132 column mode and double width characters will not work on a color monitor. Enhancements which are planned for future releases:

- Tektronix 4105 commands (already started!)
- make UniTerm work with the blitter-chip
- enhanced color version (colors for highlighted etc.)

Contents

Users Guide for UniTerm 2.0c (002)
Version 1.07

The Kermit file transfer protocol was developed by Frank da Cruz and Bill Catchings at Columbia University. Many thanks!

This program was developed with ST Pascal Plus from CCD.

UniTerm Users Guide

Simon Poole

February 4, 1992