

C*Base
v3.3
Programmer's Reference Guide

Copyright ©2003–2013 David Weinehall

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

A copy of the license is included in the chapter “Licenses”, section “The GNU General Public License” of the “C*Base v3.3 User’s Guide”.

Contents

1	Introduction	2
2	BASIC Extensions	3
3	Memory Maps	4
3.1	Brief Outline	5
3.2	Zero-page	6
3.3	\$0200-03FF	9
3.4	Interface Areas	20
3.5	KERNAL-Tables	21
3.6	Other Tables	21
4	Jump-Tables	22
4.1	Protocols (“P”-files)	22
4.2	“K/ML 2.0”	22
4.3	“K/ML 3.0”	25
4.4	“K/NML.ML” and “K/SWIFT.Dx”	27
5	BASIC-routines	28
6	KERNAL-routines	29
A	Special Algorithms Used	30
A.1	Sorting in Toplist	30
A.1.1	The Sorting Algorithm	30
A.1.2	Efficiency of the Algorithm	30

1 Introduction

This is documentation is mainly meant for those about to mess around with my code, but can also be interesting for those who are curious about the work I've done to C*Base since v3.2, and it should give a brief understanding of how this program works. Most of the information provided here can also be applied to C*Base v3.2 and prior, but some of the information is C*Base v3.3 specific. Where there are differences compared to the older version, I have *tried* to state so in the manual. Exceptions are the JUMP-tables, where I don't bother to document the old calls.

2 BASIC Extensions

To make the BASIC-code shorter and somewhat easier to understand, and to give provision for using MCI-commands from BASIC, there are two extensions made, compared to the normal BASIC v2.0. They are:

@ and <-

Both commands outputs text to both the modem and the screen, and both interprets MCI-commands while doing so. The only difference between them is that @ adds an carriage-return to the end, <- does not.

Note, that neither of these commands can handle anything else than strings. Thus, if you want to output the content of a variable, you must do something like this: `str$(br)` (this would turn the floating-point variable `br` into a string, and then output it, without adding a carriage-return.)

One thing you should keep in mind is, that these commands do not ‘tokenize’ when compiling the code in (for instance) BLITZ, so the compiled program becomes larger.

3 Memory Maps

The addresses given here are inclusive; i.e., **\$0801-9DFF** denotes the memory from **\$0801** to, and including, **\$9DFF**. The memory-areas denotes the memory reserved, not the memory actually used.

3.1 Brief Outline

Memory	Use
0801–9DFF	BASIC-memory, used by the “ C ”-files
9E00–9E27	Status-Window, Row 1
9E28–9E4F	Status-Window, Row 2
9E50–9E77	AT Init-string for Modem
9E78–9E8A	“ K/ML_0.O ” - Timing tables for the “ K/NML.ML ”-files
9E8A–9F78	“ K/ML_0.O ” - Code
9F78–B7E7	“ K/ML_1.O ”
B7E8–B7FF	Temporary Storage for Password
B800–B8FF	Buffer for the Caller’s Log; flushed to disk when full
B900–BFF7	Protocol Overlay Area
BFF8–BFFF	Protocol Interface Area; communication with the protocols is performed via these registers
C000–C5FF	Text-Prompt Memory part 2 ¹
C600–C6FF	Character Conversion Table; used for ANSI/ASCII conversion
C700–C8FF	“ K/ML_2.O ”
C900–C9FF	“ K/ML_3.O ”
CA00–CAFF	RS-232 input-buffer
CB00–CBFF	Text-Prompt Memory part 3
CC00–CDFF	NMI-handler Overlay Area
CE00–CEFF	Text input-buffer
CF00–CFFF	Text output-buffer
D000–D2FF	Access-Group Data
D300–DDFF	Text-Prompt Memory part 1
DE00–DEFF	Hardware Interface Page part 1 ²
DF00–DFFF	Hardware Interface Page part 2 ³
E000–FFFF	Text-Memory used by the Message-maker

¹If a REU is connected, **\$C500–C5FF** is used as an Interface-page, and thus the memory available for prompts will be shrunk with \$100 bytes.

²Used by Hardware such as the Ramlink, the SwiftLink/Turbo232, REU:s or similar products

³Used by Hardware such as the Ramlink, the SwiftLink/Turbo232, REU:s or similar products

3.2 Zero-page

Since at least some of these addresses are accessed from BASIC as well, the decimal form of the address has been included in this table.

Name	Memory	Description
MOCHRS	\$02 (2)	CHR\$() value for char to output
VALTYP	\$0D (13)	Variable type: String or Numeric
INDEX	\$22-25 (34-37)	Temporary workspace for the BASIC
FRETOP	\$33-34 (51-52)	Top of Free BASIC-memory
VARTAB	\$2D-2E (45-46)	Start of BASIC Variable memory
ARYTAB	\$2E-2F (47-48)	Start of BASIC Array memory
FRETOP	\$33-34 (51-52)	End of BASIC String memory
FRESPC	\$35-36 (53-54)	Temporary pointer for strings
MEMSIZ	\$37-38 (55-56)	End of the BASIC-memory
OLDLIN	\$3B-3C (59-60)	Previous BASIC line-number
OLDTXT	\$3D-3E (61-62)	Pointer to the current BASIC-statement
VARNAM	\$45-46 (69-70)	Current BASIC Variable name
VARPNT	\$47-48 (71-72)	Pointer to the current BASIC variable value
BASTMP10	\$60 (96)	BASIC Numeric work-area
FACSGN	\$66 (102)	Floating Point Accumulator 1: Sign
CHRGET	\$73 (115)	Subroutine that fetched the next BASIC-character
TXTPTR	\$7A-7B (122-123)	Pointer to the memory where CHRGET should get the char
STATUS	\$90 (144)	Status-register for Serial-communication
SVXT	\$92 (146)	Unused
XSAV	\$97 (151)	Temporary storage for the X-register
DFLTN	\$99 (153)	Device-number for Input
DFLTO	\$9A (154)	Device-number for output
OUTBLN	\$9B (155)	Length of string in output-buffer
INPLEN	\$9C (156)	Length of string in input-buffer
MSGFLG	\$9D (157)	Flag: Show Kernal error-messages
PTR1	\$9E (158)	Temporary variable used in the RS-232 handling routines
TIME	\$A0-A2 (160-162)	Software Jiffy-clock
FREKZPA3	\$A3-A4 (163-164)	Temporary work-address used heavily by the ML-files

Name	Memory	Description
G5ERROR	\$A5 (165)	Return-code from Text-Input or the Message-maker 0 - Normal execution 1 - Carrier lost 2 - Timeout 3 - User kicked off by Sysop 04-09 - N/A 0A - Read message 0B - Save message 0C - Edit line 0D - Delete lines; xx to yy 0E - Replace a word 0F - Read message, with line-numbers 10 - Message aborted 11 - View helptext 12 - Load text 13 - MCI-read 14 - Global replace 15 - Insert a line 16-27 - N/A 28 - [Ctrl + Z] (Delete line/undo input)
BUFPNT	\$A6 (166)	Unused
BITCI	\$A8 (168)	RS-232 Input bit count
RIDATA	\$AA (170)	RS-232 Start-bit check-flag
SAL	\$AC-AD (172-173)	Pointer to First buffer used by protocols
EAL	\$AE-AF (174-175)	Pointer to Second buffer used by protocols
PRMPTNR	\$B0 (176)	number of prompt to output
LINENR	\$B0-B1 (176-177)	Current Line-number in Message-maker
USERNR	\$B0-B1 (176-177)	Number of user
MSGHIMEM	\$B2-B3 (178-179)	Pointer used to address the Message-memory
BITTS	\$B4 (180)	RS-232 Output bit count
NXTBIT	\$B5 (181)	RS-232 Next bit to send

Name	Memory	Description
RODATA	\$B6 (182)	RS-232 Output byte buffer
FA	\$BA (186)	Current Device-number
FREKZPC3	\$C3-C4 (195-196)	Temporary work-address used heavily by the ML-files
LSTX	\$C5 (197)	Matrix coordinate of latest keypress (64 = None)
NDX	\$C6 (198)	Number of characters in the keyboard-buffer
RVS	\$C7 (199)	Reverse (0 = No)
SFDX	\$CB (203)	Matrix coordinate of current key pressed
PNTR	\$D3 (211)	Cursor position on logical line (0-79)
QTSW	\$D4 (212)	Editor in Quote mode (0 = No)
INSRT	\$D8 (216)	Editor in Insert mode (0 = No)
RIBUF	\$F7-F8 (247-248)	RS-232 Input-buffer pointer
ROBUF	\$F9-FA (249-250)	RS-232 Output-buffer pointer
FREKZP	\$FB-FC (251-252)	Temporary work-address used heavily by the ML-files. This address, together with FREKZP2 , also serves as an information-carrier between the BASIC-code and the ML-code
FREKZP2	\$FD-FE (253-254)	This register roughly has the same uses as FREKZP
BASZPT	\$FF (255)	BASIC temporary-register

3.3 \$0200-03FF

A lot of these addresses are accessed from BASIC as well, hence the decimal form of the address has been included in this table.

Name	Memory	Description
BUF	\$0200 (512)	The BASIC line editor input buffer
KEYD	\$0277-0280 (631-640)	Keyboard buffer
MEMSIZ	\$0283-0284 (643-644)	End of BASIC-memory
COLOR	\$0286 (646)	Text-colour
SHFLAG	\$028D (653)	Is [SHIFT]/[CTRL]/[C=] pressed?
M51CTR	\$0293 (659)	Mock 6551 control register
RSSTAT	\$0297 (663)	RS-232 Status register
BAUDOF	\$0299-029A (665-666)	Prescaler values for Baud-rate timing
RIDBE	\$029B (667)	RS-232 Pointer to end of receive-buffer
RIDBS	\$029C (668)	RS-232 Pointer to start of receive-buffer
RODBS	\$029D (669)	RS-232 Pointer to start of transmit-buffer
RODBE	\$029E (670)	RS-232 Pointer to end of transmit-buffer
ENABL	\$02A1 (673)	NMI-mask register
STORAGE	\$02A7-02BA (679-698)	Device, Drive and DOS-command info for Modules (N/A in the C*Base v3.3)
RAINCNR	\$02BB (699)	Counter for Rainbow-mode
SPRITEDEF	\$02C0-02FF (704-767)	Cursor-sprite
IGONE	\$0308-0309 (776-777)	Vector to the BASIC-token interpreter
CARRTYPE	\$0313 (787)	Carrier-type: 0 - Normal 16 - Inverted
ICHKIN	\$031E-031F (798-799)	Vector to CHKIN
IBASIN	\$0324-0325 (804-805)	Vector to BASIC-interpreter
IBSOUT	\$0326-0327 (806-807)	Vector to CHROUT
SYREBYTE	\$033C (828)	Byte that tells whether the logon was performed from the wait-screen or remotely, and to tell whether remote-mode was entered from wait-screen. Necessary for the BBS to know where it should return: 0 - Remote call 1 - N/A 2 - N/A 3 - Sysop logon 4 - Remote called from Wait-screen

Name	Memory	Description
SYSOPIN	\$033D (829)	Is the Sysop in: 0 - No 1 - Yes
STATB1	\$033E (830)	Configuration-byte; the different bits mean: 0 - Allow 300 Baud users: 0 - No 1 - Yes 1 - Save Caller's Log (C*Base v3.3): 0 - Yes 1 - No 1 - Callback validation (C*Base v3.2 and earlier): 0 - Off 1 - On 2 - Auto-feedback after application: 0 - On 1 - Off 3 - Oneliners: 0 - On 1 - Off 4 - Feedback: 0 - To Disk 1 - To Printer 5 - Caller Log: 0 - To Disk 1 - To Printer 6 - Validate disk on logoff: 0 - No 1 - Yes 7 - View disk-status: 0 - No 1 - Yes
AXSGRP	\$033F (831)	Access-Group that the user online belongs to
BRBBS	\$0340-0341 (832-833)	Maximum baudrate-setting for the BBS

Name	Memory	Description
TMCBYTE1	\$0342 (834)	C*Base v3.3 Configuration-byte 1; the different bits mean: 0 - Phone# Completion: 0 - Off 1 - On 1 - Post Checking: 0 - No 1 - Yes 2 - Carrier-Drop Checking: 0 - No 1 - Yes 3 - Military Time Format: 0 - No 1 - Yes
DISKIN	\$0343 (835)	Used to indicate whether the system-disk was inserted or not in C*Base v3.2 and prior. In the C*Base v3.3 it's only used as a file-counter when loading the Kernel-files from "C/BOOT"
NRCALLST	\$0344 (836)	Number of Calls today
NRFEEDBK	\$0345 (837)	Number of Feedback unread
MODEMTYP	\$0346 (838)	Modem-type used: 1 - Hayes 2400 Baud 2 - 1670 Regular 3 - Avatex Regular 4 - Avatex HC 5 - 1670 v2.0 6 - Hayes 1200 Baud 7 - No Modem/NULL-modem 8 - 9600 Baud/Turbo-Master Acceleration 9 - Swiftlink 2400 Baud 10 - Swiftlink 9600 Baud 11 - Swiftlink 19200 Baud 12 - Swiftlink NULL-modem
NRPOSTST	\$0347 \$ (839)	Number of Posts today

Name	Memory	Description
ONLINE	\$0348 (840)	User online: 0 - Yes 1 - No
GALLOWED	\$0349 (841)	Allowed guest-access to the BBS: 0 - No 1 - Yes
KEYBLOCK	\$034A (842)	Keyboard lockout-mode: 0 - Off 1 - On
AUTOPLNS	\$034B (843)	# of lines before Auto-Pause (0 = Off)
POSTVAL	\$034C (844)	# of credits a post is worth
NRAPPT	\$034D (845)	Number of Applications today
COLBASNG	\$034E (846)	Multi-colour Basing on (N/A in C*Base v3.3): 0 - No 1 - Yes
NROPEN	\$034F (847)	Number of different open-screens used (Default = 1)
GFXON	\$0350 (848)	PET-Graphics: 0 - Off 1 - On
BOTEMP	\$0351 (849)	Temporary variable used by BBSOUT
PAUSELIN	\$0352 (850)	Auto-pause counter
FILESUPT	\$0353 (851)	# of files uploaded today
FILESDT	\$0354 (852)	# of files downloaded today
BLKSUPT	\$0355-0356 (853-854)	Number of blocks uploaded today
MOCHRS	\$0357 (855)	See \$02 (2)
OUTBLEN	\$0358 (856)	See \$9B (155)
SCANFLG	\$0359 (857)	Scanning for Post: 0 - No 1 - Yes
NETHRS	\$035A (858)	# of hours to wait for Network (N/A in C*Base v3.3)
DATECFLG	\$035B (859)	Change Date: 0 - Yes 1 - No

Name	Memory	Description
WINCVAR	\$035C (860)	Value in Status-Window to change: 0 - None 1 - SL (Security Level) 2 - TL (Time Limit) 4 - BU (Blocks Uploaded) 8 - BD (Blocks Downloaded)
TIMOB	\$035D (861)	This address is used to check for Time-Out
SCANB	\$035D-035E (861-862)	Used when scanning for a Post
RES4	\$035F (863)	Reserved
TEMPV1	\$0360 (864)	Temporary variable #1
TEMPV2	\$0361 (865)	Temporary variable #2
BLKSDT	\$0362-0363 (866-867)	Blocks Downloaded today
TEMPI1	\$0364 (868)	Temporary IRQ-variable #1
TEMPI2	\$0365 (869)	Temporary IRQ-variable #2
PRIMEB	\$0366 (870)	Beginning of Primetime (24h format) (N/A in C*Base v3.3)
PRIMEHR	\$0367 (871)	# of Prime hours (N/A in C*Base v3.3)
KOTEMP	\$0368 (872)	Temporary storage used by the kernel-output routine
WABORT	\$0369 (873)	Allow welcome message to be aborted: 0 - Yes 1 - No
RAINBM	\$036A (874)	Rainbow-mode: 0 - Normal 1 - Char 2 - Line 3 - Word 4 - Punctuation
RAINBC1	\$036B (875)	Rainbow-Colour #1, Punct. Mode Char colour
RAINBC2	\$036C (876)	Rainbow-Colour #2, Punct. Mode Punctuation colour
RAINBC3	\$036D (877)	Rainbow-Colour #3, Punct. Mode Capital colour
RAINBC4	\$036E (878)	Rainbow-Colour #4
RAINBC5	\$036F (879)	Rainbow-Colour #5
MODTL	\$0370-0371 (880-881)	Module storage for Time-limit (N/A in C*Base v3.3)

Name	Memory	Description
D9060	\$0372 (882)	D9060-HardDrive provision activated: 0 - No 8 - Yes
UDLUNIT	\$0373 (883)	Logical Device for the U/D-directory files
BOUTFL	\$0374 (884)	Flag used by the BBS-OUT-routine
INPLEN	\$0375 (885)	See \$9C (156)
LOGBLEN	\$0376 (886)	Length of Caller's log buffer
CHATFLAG	\$0377 (887)	Chat-mode: 0 - Off 1 - On
BAUDRATE	\$0378-0379 (888-889)	Temporary storage for the Baud-rate
RES5	\$037A (890)	Reserved
RES6	\$037B (891)	Reserved
NRPOSTSC	\$037C (892)	# of posts done this call
SMAILF	\$037D (893)	Create a new file for every mail: 0 - No 1 - Yes
RMBYTE	\$037E (894)	Storage for the RM%-byte
BUSPLIT	\$037F-0381 (895-897)	Total amount of blocks uploaded (H/L/M)
BDSPLIT	\$0382-0384 (898-900)	Total amount of blocks downloaded (H/L/M)
RES7	\$0385 (901)	Reserved
SEPASCON	\$0386 (902)	Use separate ASCII-screens: 0 - No 1 - Yes
MODSUBS	\$0387 (903)	Use modular Subs (N/A in C*Base v3.3)
MODUD	\$0388 (904)	Use modular U/D-area (N/A in C*Base v3.3)
MODREM	\$0389 (905)	Use modular Remote-area (N/A in C*Base v3.3)
MDSTATUS	\$038A (906)	Module-status: 0 - Running Main-BBS 1 - Running a Module (N/A in C*Base v3.3) 2 - Exiting an module with error 3 - Running in the Application Module

Name	Memory	Description
MODTERMD	\$038B (907)	Device# for the modular Term-Prog (N/A in C*Base v3.3)
MODDEV	\$038C (908)	Device# for modules (N/A in C*Base v3.3)
LINENUM	\$038D (909)	Current line#
LCOL	\$038E (910)	Bright colour [F1]
MCOL	\$038F (911)	Medium colour [F3]
DCOL	\$0390 (912)	Dark colour [F5]
LIBDIM	\$0391 (913)	# of Libraries to dimension (N/A in C*Base v3.3)
UDDIM	\$0392 (914)	# of U/D-directories to dimension
SUBDIM	\$0393 (915)	# of Subs to dimension
ARSDIM	\$0394 (916)	Size of the Work-array AR\$() used by the user-edit; should be twice as big as the number of questions (the other half is used for the answers)
NETSTART	\$0395 (917)	Hour to start networking (N/A in C*Base v3.3)
PBEFUD	\$0396 (918)	# of posts needed before U/D-access
PBEFMOD	\$0397 (919)	# of posts needed before Module-access (N/A in C*Base v3.3)
UDTITLES	\$0398 (920)	Use U/D title-screen: 0 - No 1 - Yes
NREND	\$0399 (921)	Number of end-screens (Default = 1)
MDMHOOK	\$039A (922)	Put modem on hook during local mode: 0 - No 101 - Yes
SLOWSPD	\$039B (923)	Slow-mode: 0 - Off \$01-\$0F - On (greater number = slower)
NRONELIN	\$039C (924)	Number of one-liners
RES9	\$039D (925)	Reserved
NRDLC	\$039E (926)	# of downloads this call
MAXDLC	\$039F (927)	Max # of downloads/call
SYSSTORE	\$03A0-03B1 (928-945)	Device #, Drive # and DOS-command for System-disk

Name	Memory	Description
SSYSACT	\$03B2 (946)	Store the sysop's actions in the Caller log: 0 - No 1 - Yes
APPDEV	\$03B3 (947)	Device# for "C/APP_MOD"
MAINDEV	\$03B4 (948)	Device# for "C/BBS"
UDDEV	\$03B5 (949)	Device# for the Protocols
LTKDEV	\$03B6 (950)	LT Kernal Device# (Default = 0)
TISTEMP	\$03B7-03BD (951-957)	ti\$ temporary storage during Module-switch
CURCOL	\$03BE (958)	Cursor-colour
NPTEMP	\$03BF-03C0 (959-960)	np% temporary storage
RCHRIN	\$03C1-03C2 (961-962)	Redirection of CHRIN
RCHROUT	\$03C3-03C4 (963-964)	Redirection of CHROUT
RCHKIN	\$03C5-03C6 (965-966)	Redirection of CHKIN
RNMI	\$03C7-03C8 (967-968)	Redirection of the NMI
TOPTHRES	\$03C9 (969)	Threshold value for the Toplister file
DATEMP	\$03CA-03D3 (970-979)	da\$ temporary storage during Module-switch
QUIETM	\$03D4 (980)	Quiet-mode: 0 - Off 1 - On
MSGNET	\$03D5 (981)	Message Networking (N/A in C*Base v3.3): 0 - No 1 - Yes
ANSIFLG	\$03D6 (982)	ANSI-mode: 0 - Off 1 - On
WWCLMNS	\$03D7 (983)	# of columns for Word-Wrap
SYSOPKEY	\$03D8 (984)	CHR\$ () of key pressed by Sysop
RPFLAG	\$03D9 (985)	G5-inputmode: 0 - Input a string, with echo 1 - Input a char, with echo 2 - Input a char, without echo 4 - Password (Text is substituted with *)
PREVINPL	\$03DA (986)	Length of previous string; used by the un-erase feature
DOTMODE	\$03DB (987)	In "Dot-mode": 0 - No 1 - Yes

Name	Memory	Description
MSGMAKON	\$03DC (988)	Message-maker: 0 - Off 1 - On
MSGHIMEM	\$03DD-03DE (989-990)	See \$B2-B3 (178-179)
CURMODE	\$03E0 (992)	Cursor-mode On/Off: 0 - Off 1 - On
CASEFLG	\$03E1 (993)	UPPER/lower case flag in Message-maker: 0 - lowercase 1 - UPPERCASE
WORDWRAP	\$03E2 (994)	Word-Wrap: 0 - Off 1 - On
ABONOFF	\$03E3 (995)	Allow messages to be aborted: 0 - No 1 - Yes
G5ERROR	\$03E4 (996)	See \$A5 (165)
PRMPTNR	\$03E5 (997)	See \$B0 (176)
LINENR	\$03E5-0358 (997-998)	See \$B0-B1 (176-177)
LINENR	\$03E5-0358 (997-998)	See \$B0-B1 (176-177)
USERNR	\$03E5-0358 (997-998)	See \$B0-B1 (176-177)
TIMEUNIT	\$03E7 (999)	Time before answering the phone
TEMPAG	\$03E8-03FC (1000-1020)	Temporary storage for user-group access

Name	Memory	Description
AGBITS	\$03E8 (1000)	Access-bits: 0 - Library Access (N/A in C*Base v3.3): 0 - No 1 - Yes 1 - Mail Access: 0 - No 1 - Yes 2 - Prime-time Access (N/A in C*Base v3.3): 0 - No 1 - Yes 3 - Module-Access (N/A in C*Base v3.3): 0 - No 1 - Yes 4 - Subs-Access: 0 - No 1 - Yes 5 - Caller Log-Access: 0 - No 1 - Yes 6 - BBS list-Access: 0 - No 1 - Yes 7 - Back to Back calls: 0 - No 1 - Yes
AGRES1	\$03E9 (1001)	Reserved
AGRES2	\$03EA (1002)	Reserved
AGRES3	\$03EB (1003)	Reserved
AGRES4	\$03EC (1004)	Reserved

Name	Memory	Description
AGBITS2	\$03ED (1005)	Access-bits: 0 - 300 Baud Access to U/D-areas: 0 - No 1 - Yes 1 - 1200 Baud Access to U/D-areas: 0 - No 1 - Yes 2 - 2400 (and above) Baud Access to U/D-areas: 0 - No 1 - Yes 3 - Reserved 4 - Reserved 5 - Reserved 6 - Reserved 7 - Reserved
OMNIAXS	\$03EE (1006)	Access to write OMNI-messages: 0 - No 1 - Yes
ONELNAXS	\$03EF (1007)	Access to write One-liners: 0 - No 1 - Yes
MCIAXS	\$03F0 (1008)	Access to MCI-commands: 0 - No 1 - Yes
CALLSDAY	\$03F1 (1009)	Calls allowed/Day (0 = Unlimited)
AMTIMLIM	\$03F2 (1010)	AM Time Limit (0 = Unlimited)
PMTIMLIM	\$03F3 (1011)	PM Time Limit (0 = Unlimited)
POSTNEED	\$03F4 (1012)	Must follow minimum post-limit: 0 - Yes 1 - No
MODCREDS	\$03F5 (1013)	Credits/module (N/A in C*Base v3.3)
UDRATIO	\$03F6 (1014)	Upload/Download ratio (0 = Unlimited Credits)
AGRES5	\$03F7 (1015)	Reserved
AGRES6	\$03F8 (1016)	Reserved
AGRES7	\$03F9 (1017)	Reserved
AGRES8	\$03FA (1018)	Reserved
AGRES9	\$03FB (1019)	Reserved
AGRES10	\$03FC (1020)	Reserved

3.4 Interface Areas

These memory-locations are meant for communication between the BASIC-files, and between the BASIC and the ML-files.

Name	Memory	Description
XTRNPAGE	\$07E8-07F7 (2024-2039)	Interface area for communication with other files
NUTMP	\$07E8-07E9 (2024-2025)	Temporary storage used to transfer the nu%-variable to "C/TOPLISTER"
MIDBLNK	\$07EA (2026)	Flag to tell "C/TOPLISTER" whether it was called manually or not: 0 - Manually 1 - From the Midnight Routine
	\$07EB-07F7 (2027-2039)	Reserved
PROTPAGE	\$BFF8-BFFF (49144-49151)	Protocol interface area
PNTBLKS2	\$BFF8 (49144)	Block-size of Punter blocks
PROTFLG1	\$BFF9 (49145)	Not sure; some kind of flag ???
PROTRES0	\$BFFA (49146)	Reserved
FILETYPE	\$BFFB (49147)	File-type: 1 - PRG 2 - SEQ
PROTRES1	\$BFFC (49148)	Reserved
PROTRES2	\$BFFD (49149)	Reserved
PROTRES3	\$BFFE (49150)	Reserved
PROTID	\$BFFF (49151)	Reserved

3.5 KERNAL-Tables

These are tables within the KERNAL-memory that the BBS makes use of. An “@”-sign denotes that a non-standard name is used.

Name	Memory	Description
PETCOLS	\$E8DA (59610)	@; Conversion table for all the PET-colours to/from numerical value from/to PET-ASCII value
KDECODE	\$EB81 (60289)	@; Keyboard decode table

3.6 Other Tables

These are tables within the “K”-files that the BBS makes use of.

Name	Memory	Description
TIMTAB1	\$9E78 (40568)	Timing-Data used by the “K/NMI.ML”-files ⁴
TIMTAB2	\$9E8A (40578)	Timing-Data used by the “K/NMI.ML”-files ⁵

⁴This table was in the “ML_2.O”-file in C*Base v3.2 and prior

⁵This table was in the “ML_2.O”-file in C*Base v3.2 and prior

4 Jump-Tables

Various calls into the assembler-code.

4.1 Protocols (“P”-files)

Name	Memory	Description
RECEIVE	\$B900 (47360)	Receive a file
SEND	\$B903 (47363)	Send a file
INITRECV	\$B906 (47366)	Initialise Receive
INITSEND	\$B909 (47369)	Initialise Send
PROTINIT	\$B90C (47372)	Initialise Protocol
SETPRID	\$B90F (47375)	Set Protocol Identification Byte

4.2 “K/ML 2.0”

Name	Memory	Description
READ2SCR	\$C700 (50944)	Output a seq-file opened as channel #8. No modem-output and no MCI-interpreting
CARRCHK	\$C708 (50952)	Jump to the routine that checks for carrier loss. FREKZP > 0 means that the carrier is lost
CLRCHN2	\$C710 (50960)	Clear all channels
ACTTERM	\$C718 (50968)	Activate the Mini-Term
READWLEN	\$C720 (50976)	Read a line with the exact length available in BASZPT from the seq-file opened as channel #8. The string is put into the output-buffer
GETLMEM	\$C728 (50984)	Call with a line# in FREKZP (W). This routine will return a memory-pointer to that line in FREKZP (W)
ACTRDISK	\$C730 (50992)	Activate the Ram-disk used to store “ STATS ” temporarily while in the “ C/APP_MOD ”

Name	Memory	Description
ACTPROT	\$C738 (51000)	Initialise the loaded Protocol
PROT1	\$C740 (51008)	Initialise Receive
PROT2	\$C748 (51016)	Receive File
PROT3	\$C750 (51024)	Initialise Send
PROT4	\$C758 (51032)	Send File
SETPRID	\$C760 (51040)	Set Protocol Identification Byte
PRMPTOUT	\$C768 (51048)	Output a prompt corresponding to the value in PRMPTNR
BLINPRT	\$C770 (51056)	Redirected jump to LINPRT
FREESTR	\$C778 (51064)	Redirected jump to FRESTR
FOUT2	\$C780 (51072)	Redirected jump to FOUT
MOVFM2	\$C788 (51080)	Redirected jump to MOVFM
BGIVAYF	\$C790 (51088)	Redirected jump to GIVAYF
GETABYTE	\$C798 (51096)	Get a byte. The more correct call is READLB
CREATEIS	\$C7A0 (51104)	Create a BASIC-string from the text-line in the output-buffer, and assign it the string-handle $i\$$
GETNRLIN	\$C7A8 (51112)	After a call, LINENR (W) will contain the number of lines in the message
NEWDAY	\$C7B0 (51120)	Routine called every new day; resets daily values
DIRQSOFF	\$C7B8 (51128)	Turn off the Window-IRQ
DIRQSON	\$C7BB (51131)	Turn on the Window-IRQ
DASC2PET	\$C7BE (51134)	Convert the content of the input-buffer from ASCII to PETSCII
CHKCARR	\$C7C1 (51137)	Check for Carrier loss
INIT1ST	\$C7C4 (51140)	Initialise some values before running the different “C”-files
SCR2WIN	\$C7CC (51148)	Move User-data from BASIC to the information Window. The information is taken from the first row of the screen, and contains the user’s handle with the colours stripped. This is a crude solution, but it works, and takes far less memory than any other. Still, I hope to come up with something better in a later version. . .

Name	Memory	Description
MCION	\$C7D4 (51156)	Turn MCI-commands ON
MCIOFF	\$C7D7 (51159)	Turn MCI-commands OFF
ATBAS	\$C7DD (51165)	Routine for the BASIC-extensions; @ and <-
RIGONE	\$C7F2 (51186)	Redirection of IGONE
BLANK2MH	\$C864 (51300)	Blank the Screen; if a C= 128 is used and no SuperCPU is present, turn on 2 MHz mode
BLANK	\$C873 (51315)	Blank the screen
SLOWMODE	\$C87C (51324)	Turn off 2 MHz mode and turn the screen back on. SuperCPU's are NOT affected
NO2MHZ	\$C884 (51332)	Turn off 2 MHz mode
SCPUON	\$C88D (51341)	Detect the CMD SuperCPU and, if present, activate it
FORMEVL	\$C881 (51361)	Redirected jump to FRMEVL
SCPUOFF	\$C8A9 (51369)	Detect the CMD SuperCPU and, if present, deactivate it

4.3 “K/ML 3.0”

Name	Memory	Description
ATPRINT	\$C900 (51456)	Print the char available in MOCHRS (CHR\$() -value) to the screen and modem using the @ MCI-command
SETCUR	\$C908 (51464)	Position the Sprite-cursor where the char-cursor is
IRQSOFF	\$C910 (51472)	Jump to IRQSOFF2
IRQSON	\$C918 (51480)	Jump to IRQSON2
READLB	\$C920 (51488)	Read one byte from channel# stored in FREKZP + 1 . This byte is stored in FREKZP
ASC2PET	\$C928 (51496)	Convert the CHR\$() of FREKZP to a PETSCII-value, given that the value in FREKZP was an ASCII/ANSI-value
READLN	\$C930 (51504)	Read a line from channel #8. This will read until a Carriage Return (#\$0D), or 255 bytes been read. The line is stored in the output-buffer
KL2OUTB	\$C938 (51512)	Copy a string from the memory below the kernal to the output-buffer. Memory-address is decided by the content of FREKZPC3 (W)

Name	Memory	Description
MAKEIS	\$C940 (51520)	Create i\$ from the text found in the input-buffer
SETVAR	\$C948 (51528)	Jump to SETUPVAR
G5INP	\$C950 (51536)	General input-routine; inputs a char, a line or an entire text
GETSPACE	\$C958 (51544)	Jump to GETSPAR
CHRIN2	\$C960 (51552)	Jump to CHRIN
BUFSWAP	\$C968 (51560)	Swap the contents of the input-buffer and the output-buffer with each other
BOING	\$C970 (51568)	Sound a nice little “boing”
WIRQON	\$C978 (51576)	Turn on the IRQ for the information-window
RSTIRQ	\$C980 (51584)	Reset the IRQ-jump to point to the normal IRQ
MOVMEM	\$C988 (51592)	FREKZPC3 (W) = beginning of memory area to move, FREKZP (W) = end of memory area to move. The difference between INPLEN and OUTBLEN will decide how many bytes to move up or down in memory (0-255)
SEARCH	\$C990 (51600)	Search the message-memory for the specified string. The string to search for should be in the input-buffer and the length of the string in OUTBLEN
RDIRL	\$C998 (51608)	Read a line from the directory to the output-buffer, and splits it up to its different parts; name, type and blksize
SETBASP	\$C9A0 (51616)	Set the pointer to the search-string; use this before calling SEARCH . The pointer is taken from FREKZPC3 (W)
BUF2MEM	\$C9A8 (51624)	Copy the output-buffer to the address specified at FREKZPC3 (W)
DELCOPY	\$C9B0 (51632)	Delete a message/post or copy a file
SAVEMSG	\$C9B8 (51640)	Save a message written in the message-maker, starting at the address in FREKZPC3 (W) and ending at the address in FREKZP (W)

Name	Memory	Description
BUFOUT	\$C9C0 (51648)	Output the contents of the output-buffer to screen and Modem, with MCI-translation
DSK2SM	\$C9C8 (51656)	Read a file from channel #8 to screen + modem
MSG2SM	\$C9D0 (51664)	Read a post from channel #8 to screen + modem. If you press A to abort or S to skip, it scans to the next # \$FF (message separator)
FFSCAN	\$C9D8 (51672)	Scan for a message separator (# \$FF)
DELBCHR	\$C9E0 (51680)	Delete a char in the input-buffer
GETNROLN	\$C9E8 (51688)	Multi-use routine, used by the message-maker. This routine can do the following things: <ul style="list-style-type: none"> • Count number of lines in message • Show the message • Show the message with MCI-interpretation • Show the message with line-numbers • Prepare for editing the last line in the message
RAMDISK	\$C9F0 (51696)	Jump to the RAMDISK
RESETIO	\$C9F8 (51704)	Reset the IO-vectors

4.4 “K/NMI.ML” and “K/SWIFT.Dx”

Name	Memory	Description
NMION	\$CC00 (52224)	Turn on the NMI used to communicate with the modem
CHKCARR	\$CC03 (52227)	Check for carrier-loss
DTROFF	\$CC06 (52230)	Turn DTR off
DTRON	\$CC09 (52233)	Turn DTR on
GOTO232	\$CC0C (52236)	???
STOP232	\$CC0F (52239)	Turn off the NMI used to communicate with the modem BASIC-routines

5 BASIC-routines

An @-sign denotes that a non-standard name and/or entry-point is used.

Name	Memory	Description
NEWSTT3	\$A7BE (42942)	@; Skips STOPCHK and a NOP, thus a little faster than NEWSTT . Set up next statement for execution
GONE2	\$A7EF (42991)	@; Faster than GONE . Read and execute next statement.
FRMEVL	\$AD9E (44446)	Evaluate expression
SETUPVAR	\$B0E7 (45287)	@; no check for existence, faster than PTRGET . Set up a Variable
GIVAYF	\$B391 (45969)	Convert 16-bit signed integer to floating point
GETSPA	\$B4F4 (46324)	Allocate space in memory for string
GARBAG	\$B526 (46374)	Collect garbage; remove unused string-data
FRESTR	\$B6A3 (46755)	Discard a temporary string
MOVFM	\$BBA2 (48034)	Move a floating point# from memory to FAC2
LINPRT	\$BDCD (48589)	Output a number in ASCII-decimal form Input high-byte in A and low-byte in X
FOUT	\$BDDD (48605)	Convert FAC1 to an ASCII-string A is set to the high-byte, and X as the low-byte pointing at the string

6 KERNAL-routines

An @-sign denotes that a non-standard name and/or entry-point is used.

Name	Memory	Description
CHROUT2	\$E716 (59158)	@; output a char to the opened channel
GETIN2	\$F142 (61762)	@; input a char from the opened channel
READST	\$FFB7 (65463)	Read I/O status
CLOSE	\$FFC3 (65475)	Close file
CHKIN	\$FFC6 (65478)	Change standard input-channel
CHKOUT	\$FFC9 (65481)	Change standard output-channel
CLRCHN	\$FFCC (65484)	Clear channel
CHRIN	\$FFCF (65487)	Input a char from the opened channel
CHROUT	\$FFD2 (65490)	Output a char to the opened channel
GETIN	\$FFE4 (65493)	Input a char from the keyboard
PLOT	\$FFF0 (65520)	Read or change cursor-position

A Special Algorithms Used

A.1 Sorting in Toplister

The sorting done in “C/TOPLISTER” is performed with an improved bubble-sort algorithm. The difference from normal bubble-sort is that using this algorithm you only have to move the smallest element once every iteration, and each iteration means one less element to compare with.

A.1.1 The Sorting Algorithm

- 1 Let $A_0 \cdots A_n$ be the numbers to sort
- 2 Let $B_0 \cdots B_n$ be the numbers $0 \cdots n$
- 3 Let $C = 0$
- 4 let $D = C, E = C$
- 5 Let $D = D + 1$
- 6 If element $A_D < A_C$ let $E = D$
- 7 Repeat from 5 until $D = n$
- 8 $Swap(A_C, A_E)$
- 9 $Swap(B_C, B_E)$
- 10 $C = C + 1$
- 11 Repeat from 4 until $C = n - 1$

$A_0 \cdots A_n$ will now contain all the numbers in sorted order, and $B_0 \cdots B_n$ will be a cross-reference list that can be used to get the original position of the value.

A.1.2 Efficiency of the Algorithm

The efficiency of this algorithm is $O(n^2)$ compares + $O(n)$ swaps.