Table of Contents

Introduction . . . 1
- What is Let's C? . . . 1
- Hardware requirements . . . 1
- Changes in release 4.0 . . . 1
- How to use this manual . . . 2
- User registration and reaction report . . . 3
- Technical support . . . 3
- Bibliography . . . 3
  - i8086/MS-DOS information . . . 5

Installing and Running Let's C . . . 7
- Installing Let's C . . . 7
  - Installing Let's C onto a hard disk . . . 7
  - Installing Let's C onto a floppy-disk system . . . 8
  - Re-installing a portion of Let´s C . . . 10
- Setting your computer´s environment . . . 10
  - Setting the PATH . . . 10
  - Finding the ccargs file . . . 11
  - Editing ccargs . . . 11
- Using MWS, the Let's C command interface . . . 12
  - Editing a file . . . 13
  - Simple compiling . . . 15
  - Running a program . . . 18
  - Quick DOS and !DOS options . . . 19
  - Using the make programming discipline . . . 19
  - Using csd, the C source debugger . . . 21
  - Resetting the buffers . . . 22
  - Where to go from here . . . 25

C for Beginners . . . 27
- Programming languages and C . . . 27
  - Assembly and high-level languages . . . 27
  - So, what is C? . . . 28
  - Structured programming . . . 28
  - Compiling a C program . . . 29
  - Writing a C program . . . 29
- A sample C programming session . . . 30
  - Designing a program . . . 30
  - The main function . . . 31
  - Opening a file and showing text . . . 32
  - Accepting file names . . . 34
  - Error checking . . . 35
  - Print a portion of a file . . . 37
  - Checking for the end of file . . . 39
  - Polling the keyboard . . . 40
  - For more information . . . 42
- Where to go from here . . . 42

Compiling with Let's C . . . 43
- The phases of compilation . . . 43
- Edit errors automatically . . . 43
- Renaming executable files . . . 44

Floating-point numbers . . . 44
Compiling multiple source files . . . 45
Wildcards. . . . 45
Tailoring the command line interface . . . 46
Linking without compiling. . . . 46
Compiling without linking. . . . 47
Mini-make option . . . 47
Assembly-language files . . . 47
Changing the size of the stack . . . 48
i8086 memory models . . . 48
Debugging information. . . . 49
i8087 programs . . . 49
Options passed to MS-LINK . . . 49
Compiling programs without STDIO . . . 50
Using default options. . . . 50
Where to go from here . . . 51

Introduction to MicroEMACS . . . 53
What is MicroEMACS? . . . 53
Keystrokes — <ctrl>, <esc> . . . 53
Becoming acquainted with MicroEMACS . . . 53
Beginning a document . . . 54
Moving the Cursor . . . 55
Moving the cursor forward. . . . 56
Moving the cursor backward . . . 56
From line to line. . . . 56
Moving up and down by a screenful of text . . . 57
Moving to beginning or end of text . . . 57
Saving text and quitting . . . 57
Killing and deleting . . . 57
Deleting versus killing . . . 58
Erasing text to the right . . . 58
Erasing text to the left . . . 59
Erasing lines of text . . . 59
Yanking back (restoring) text . . . 59
Quitting . . . 59
Block killing and moving text . . . 60
Moving one line of text . . . 60
Multiple copying of killed text. . . . 60
Kill and move a block of text . . . 60
Capitalization and other tools. . . . 61
Capitalization and lowercasing . . . 61
Transpose characters. . . . 62
Screen redraw . . . 62
Return indent . . . 62
Word wrap . . . 63
Search and Reverse Search . . . 64
Search forward . . . 64
Reverse search . . . 65
Cancel a command . . . 65
Search and replace . . . 66
Saving text and exiting. . . . 66
Write text to a new file . . . 67
Save text and exit . . . 67

Advanced editing . . . 67
Arguments . . . 68
Arguments — default values . . . 68
Selecting values . . . 69
Deleting with arguments—an exception . . . 69
Buffers and files . . . 69
Definitions . . . 69
File and buffer commands . . . 70
Write and rename commands . . . 70
Replace text in a buffer . . . 70
Visiting another buffer . . . 71
Move text from one buffer to another . . . 71
Checking buffer status . . . 71
Renaming a buffer . . . 72
Delete a buffer . . . 72
Windows . . . 72
Creating windows and moving between them . . . 73
Enlarging and shrinking windows . . . 74
Displaying text within a window . . . 75
One buffer . . . 75
Multiple buffers . . . 76
Moving and copying text among buffers . . . 76
Checking buffer status . . . 76
Saving text from windows . . . 76
Keyboard macros . . . 77
Keyboard macro commands . . . 77
Replacing a macro . . . 77
Sending commands to MS-DOS . . . 77
Compiling and debugging through MicroEMACS . . . 78
The MicroEMACS help facility . . . 79
Where to go from here . . . 79
make Programming Discipline . . . 81
How does make work? . . . 81
Try make . . . 82
Essential make . . . 83
The makefile . . . 83
Building a simple makefile . . . 84
Comments and macros. . . . 84
Setting the time . . . 85
Building a large program . . . 85
Command line options . . . 86
Other command line features . . . 86
Advanced make . . . 87
Default rules . . . 87
Double-colon target lines . . . 88
Alternative uses . . . 89
Special targets . . . 90
Errors . . . 90
Exit status . . . 90
Where to go from here . . . 90
Questions and Answers . . . 91
Programming problems . . . 91
Problems with running programs . . . 95

Limitations in i8086		96
Error Messages		97
The Lexicon		115
example	Give an example of Mark Williams Lexicon format	117
!	Logical negation operator	118
!=	Inequality operator	118
”	String literal character	118
#	String-ize operator	119
##	Token-pasting operator	120
#define	Define an identifier as a macro	121
#elif	Include code conditionally	123
#else	Include code conditionally	123
#endif	End conditional inclusion of code	124
#error	Error directive	124
#if	Include code conditionally	124
#ifdef	Include code conditionally	125
#ifndef	Include code conditionally	125
#include	Read another file and include it	126
#line	Reset line number	127
#pragma	Perform implementation-defined task	127
#undef	Undefine a macro	128
%	Remainder operator	128
%=	Remainder assignment operator	129
&		129
&&	Logical AND operator	130
&=	Bitwise-AND assignment operator	130
()		130
*		131
*/		132
*=	Multiplication assignment operator	132
+		132
++	Increment operator	133
+=	Addition assignment operator	134
,		134
-		135
--	Decrement operator	136
-=	Subtraction assignment operator	136
->	Select a member	136
.	Member selection	137
/	Division operator	138
/*		138
/=	Division assignment operator	138
:		139
;		139
<	Less-than operator	139
<<	Bitwise left-shift operator	139
<<=	Bitwise left-shift assignment operator	140
<=	Less-than or equal-to operator	140
=	Assignment operator	141
==	Equality operator	141
>	Greater-than operator	142
>=	Greater-than or equal-to operator	142
>>	Bitwise right-shift operator	143

>>= . . . . Bitwise right-shift assignment operator 144
?: . . . . Conditional operator 144
[] . . . . Array subscript operator 145
^ . . . . Bitwise exclusive OR operator 146
^= . . . . Bitwise exclusive-OR assignment operator 147
_ _DATE_ _ . . . . Date of translation 147
__end 147
_ _FILE_ _ . . . . Source file name 148
_ _LINE_ _ . . . . Current line within a source file 148
_ _STDC_ _ . . . . Mark a conforming translator 148
_ _TIME_ _ . . . . Time source file is translated 149
_exit() . . . . Terminate a program 149
_tolower() . . . . Convert letter to lower case 149
_toupper() . . . . Convert letter to upper case 150
_zero() . . . . Zero a block of memory 151
{} 151
| . . . . Bitwise inclusive OR operator 151
|= . . . . Bitwise inclusive-OR assignment operator 152
| | . . . . Logical OR operator 152
~ . . . . Bitwise complement operator 153
abort() . . . . End program immediately 154
abs() . . . . Compute the absolute value of an integer 154
access() . . . . Check if a file can be accessed in a given mode 155
access.h . . . . Define manifest constants used by access() 156
access checking 157
acos() . . . . Calculate inverse cosine 157
address 157
alias 158
alien . . . . Name a non-standard function 158
alignment 159
arena 159
argc 160
argument 160
argv 160
array declarators 161
as . . . . i8086 assembler 161
ASCII 177
asctime() . . . . Convert broken-down time to text 180
asin() . . . . Calculate inverse sine 181
assert() . . . . Check assertion at run time 181
assert.h . . . . Header for assertions 182
atan() . . . . Calculate inverse tangent 182
atan2() . . . . Calculate inverse tangent 183
atexit() . . . . Register a function to be performed at exit 183
atof() . . . . Convert string to floating-point number 184
atoi() . . . . Convert string to integer 185
atol() . . . . Convert string to long integer 185
auto . . . . Automatic storage duration 186
aux . . . . Logical device for serial port 186
behavior 187
BIOS 187
bios.h . . . . Outline ROM BIOS data area 188
bit 188

bit-fields		188
bit map		189
block		189
break	Exit unconditionally from loop or switch	190
bsearch()	Search an array	190
byte		192
byte ordering	Describe order of bytes	192
cabs()	Complex absolute value function	194
calloc()	Allocate and clear dynamic memory	194
case	Mark entry in switch table	194
cc	Compiler controller	195
cc0		200
cc1		200
cc2		200
cc3		200
CCTAIL	Variables at end of compilation command	201
ceil()	Integral ceiling	201
char		201
character constant		202
character display semantics		202
character handling		203
clearerr()	Clear a stream´s error indicator	204
CLK_TCK		205
clock()	Get processor time used	205
clock_t	System time	206
close()	Close a file	206
cmp	Compare bytes of two files	207
commands		207
comment		208
compatible types		208
compile		209
compliance		209
con	Logical device for the console	210
const	Qualify an identifier as not modifiable	210
constant expressions		210
constants		212
continue	Force next iteration of a loop	212
conversions		213
cos()	Calculate cosine	215
cosh()	Calculate hyperbolic cosine	215
cpp	C preprocessor	215
creat()	Create/truncate a file	216
csreg()	Get value from CS register	216
ctime()	Convert calendar time to text	217
ctype.h	Header for character-handling functions	218
daemon		219
date and time		219
dayspermonth()	Return number of days in a given month	220
DBL_DIG		220
decimal-point character		220
declarations		221
declarators		221
default	Default entry in switch table	222

defined . . . . Check if identifier is defined. 222
definition 223
Definitions 223
diagnostics 225
difftime() . . . . Calculate difference between two times 225
digit 226
directory 226
div() . . . . Perform integer division 226
div_t . . . . Type returned by **div()** 227
do . . . . Loop construct 227
dos.h . . . . Define MS-DOS functions and devices 228
DOS-specific features 228
double 228
dsreg() . . . . Get value from DS segment register 229
dup() . . . . Duplicate a file descriptor 229
dup2() . . . . Duplicate a file descriptor 229
ecvt() . . . . Convert floating-point numbers to strings 231
egrep . . . . Extended pattern search. 231
else . . . . Conditionally execute a statement 233
enum . . . . Enumerated data type 234
enumeration constant 235
environmental variable 235
envp . . . . Argument passed to main 235
EOF . . . . Indicate end of a file 236
errno . . . . External integer that holds error status 236
errno.h . . . . Define errno and error codes 237
escape sequences 237
esreg() . . . . Get value from ES segment register 238
exargs() . . . . Get and parse a command line 238
exception 240
execall() . . . . Execute a subprogram 240
executable file 241
exit() . . . . Terminate a program gracefully. 241
explicit conversion 242
extended character handling 243
extended time 243
extern . . . . External linkage. 244
external definitions 244
external name 244
fabs() . . . . Compute absolute value 246
false 246
fclose() . . . . Close a stream. 246
fcvt() . . . . Convert floating-point numbers to strings 247
fdopen() . . . . Open a stream for standard I/O 247
feof() . . . . Examine a stream´s end-of-file indicator 248
ferror() . . . . Examine a stream´s error indicator 249
fflush() . . . . Flush output stream´s buffer 250
fgetc() . . . . Read a character from a stream 250
fgetpos() . . . . Get value of file-position indicator 251
fgets() . . . . Read a line from a stream 252
fgetw() . . . . Read integer from stream 253
field 254
file 255

file descriptor . . . 256
FILENAME_MAX . . . Maximum length of file name . . . 256
fileno() . . . Get file descriptor . . . 256
float . . . 257
float.h . . . 260
floating constant . . . 262
floor() . . . Numeric floor . . . 262
fmod . . . Calculate modulus for floating-point number . . . 263
fopen() . . . Open a stream for standard I/O . . . 263
for . . . Loop construct . . . 265
fpos_t . . . Encode current position in a file . . . 265
fprintf() . . . Print formatted text into a stream . . . 266
fputc() . . . Write a character into a stream. . . . 267
fputs() . . . Write a string into a stream . . . 268
fputw() . . . Write an integer to a stream. . . . 268
fread() . . . Read data from a stream. . . . 268
free() . . . Deallocate dynamic memory . . . 269
freopen() . . . Re-open a stream . . . 270
frexp() . . . Fracture floating-point number. . . . 271
fscanf() . . . Read and interpret text from a stream. . . . 271
fseek() . . . Set file-position indicator . . . 273
fsetpos() . . . Set file-position indicator . . . 274
ftell() . . . Get value of file-position indicator . . . 275
function . . . 275
function call . . . 276
function declarators . . . 282
function definition . . . 282
function designator . . . 283
function prototype . . . 283
fwrite() . . . Write data into a stream . . . 285
gcvt() . . . Convert floating-point numbers to strings . . . 286
general utilities . . . 286
getc() . . . Read a character from a stream . . . 287
getchar() . . . Read a character from the standard input stream. . . . 287
getenv() . . . Read environmental variable . . . 288
gets() . . . Read a string from the standard input stream . . . 289
getw() . . . Read word from file stream . . . 290
gmtime() . . . Convert calendar time to universal coordinated time . . . 290
goto . . . Unconditionally jump within a function. . . . 291
header . . . 293
header names . . . 294
hypot() . . . Compute hypotenuse of right triangle . . . 294
i8086 support . . . 295
i8087 . . . Floating-point co-processor . . . 295
identifiers . . . 296
if . . . Conditionally execute an expression. . . . 297
implicit conversions . . . 298
inb() . . . Read from a port . . . 298
INCDIR . . . Directory that holds include files . . . 298
index() . . . Find a character in a string . . . 299
initialization . . . 299
int . . . 302
intcall() . . . Call MS-DOS interrupt. . . . 303

integer constant		304
internal name		305
interrupt		305
isalnum()	Check if a character is a numeral or letter	305
isalpha()	Check if a character is a letter	306
isascii()	Check if a character is an ASCII character	306
iscntrl()	Check if a character is a control character	306
isdigit()	Check if a character is a numeral	307
isgraph()	Check if a character is printable	307
islower()	Check if a character is a lower-case letter	307
isprint()	Check if a character is printable	308
ispunct()	Check if a character is a punctuation mark	308
isspace()	Check if character is white space	309
isupper()	Check if a character is an upper-case letter	309
isxdigit()	Check if a character is a hexadecimal numeral	310
j0()	Compute Bessel function	311
j1()	Compute Bessel function	312
jday_to_time()	Convert Julian date to system time	312
jday_to_tm()	Convert Julian date to system calendar format	312
jmp_buf	Type used with non-local jumps	313
jn()	Compute Bessel function	313
keywords		315
label		316
labs()	Compute the absolute value of a long integer	316
Language		316
LARGE model	Intel multi-segment memory model	319
LC_ALL	All locale information	319
LC_COLLATE	Locale collation information	320
LC_CTYPE	Locale character-handling information	321
LC_MONETARY	Locale monetary information	321
LC_NUMERIC	Locale numeric information	321
LC_TIME	Locale time information	322
lconv	Hold monetary conversion information	322
ldexp()	Load floating-point number	324
ldiv()	Perform long integer division	325
ldiv_t	Type returned by ldiv()	325
lexical elements		326
Lexicon		326
libcxs87.lib	Standard library, SMALL model/i8087 only	327
libm		327
LIBPATH	Directories that hold libraries	327
limits.h		328
link		329
linkage		329
locale.h	Localization functions and macros	330
localeconv()	Initialize **lconv** structure	331
localization		331
localtime()	Convert calendar time to local time	334
log()	Compute natural logarithm	335
log10()	Compute common logarithm	336
long double		336
long int		337
longjmp()	Execute a non-local jump	337

lseek()	Set read/write position.	338
lvalue		339
main		341
main	Introduce program´s main function	341
make	Program building discipline	342
malloc()	Allocate dynamic memory	345
manifest constant		345
math.h	Header for mathematics functions	346
mathematics		346
maxmem		347
mblen()	Return length of a string of multibyte characters	347
mbstowcs()	Convert sequence of multibyte characters to wide characters	348
mbtowc()	Convert a multibyte character to a wide character	348
me	MicroEMACS screen editor	349
member		355
memchr()	Search a region of memory for a character	356
memcmp()	Compare two regions	357
memcpy()	Copy one region of memory into another	358
memmove()	Copy region of memory into area it overlaps	359
memset()	Fill an area with a character	360
mktemp()	Generate a temporary file name	360
mktime()	Turn broken-down time into calendar time	361
model		362
modf()	Separate floating-point number.	362
mtype.h	List processor code numbers	363
multibyte characters		363
name space		366
nested comments		367
nm	Print a program´s symbol table	367
nondigit		368
non-local jumps		368
notmem()	Check if memory is allocated	369
null directive	Directive that does nothing	369
null pointer constant		369
null statement		370
numerical limits		370
nybble		370
object		371
object definition		371
object format		372
object types		372
obsolescent		372
open()	Open a file	372
operating system devices	Logical devices for system peripherals	374
operators		374
ordinary identifier		375
outb()	Write to a port	376
parameter		377
PATH	Directories that hold executable files	377
path()	Build a path name for a file	377
path.h	Declare path()	378
pattern		379
peek()	Extract a word from memory	379

peekb() . . . . Extract a byte from memory. 379
perror() . . . . Write error message into standard error stream 380
picture() . . . . Format numbers under mask. 381
pnmatch() . . . . Match string pattern 382
pointer 383
pointer declarators 386
poke() . . . . Insert a word into memory 386
pokeb() . . . . Insert a byte into memory 387
port 387
portability 387
pow() . . . . Raise one number to the power of another 388
pr . . . . Paginate and print files 389
preprocessing numbers 389
printf() . . . . Format and print text into the standard output stream 390
prn . . . . MS-DOS logical device for parallel port 396
process 397
program startup. 397
program termination 397
pun 397
punctuators 398
putc() . . . . Write a character into a stream. 398
putchar() . . . . Write a character into the standard output stream 399
puts() . . . . Write a string into the standard output stream 400
putw(). . . . . Write word to stream. 400
qsort(). . . . . Sort an array 402
raise(). . . . . Send a signal 403
rand() . . . . Generate pseudo-random numbers 404
random access. 405
read() . . . . Read from a file 405
read-only memory. 406
realloc(). . . . . Reallocate dynamic memory. 406
record. 407
register . . . . Quick access required 407
register 408
remove() . . . . Remove a file. 408
rename() . . . . Rename a file 409
return. . . . . Return to calling function 409
rewind(). . . . . Reset file-position indicator 410
rindex() . . . . Find a character in a string 411
runtime startup 411
rvalue 412
sbrk() . . . . Increase a program´s data space 413
scanf(). . . . . Read and interpret text from standard input stream 413
scope 416
sequence point 418
setbuf() . . . . Set alternative stream buffer 418
setjmp(). . . . . Save environment for non-local jump 419
setjmp.h . . . . Declarations for non-local jump 419
setlocale(). . . . . Set or query a program´s locale. 420
setvbuf() . . . . Set alternative stream buffer 421
shellsort(). . . . . Sort arrays in memory 422
short int 422
side effect. 423

sig_atomic_t . . . . Type that can be updated despite signals 423
signal() . . . . Set processing for a signal. 423
signal.h . . . . Signal-handling routines 424
signal handling 425
signals/interrupts 426
signed. 431
signed char 432
sin() . . . . Calculate sine 432
sinh() . . . . Calculate hyperbolic sine 433
size . . . . Print the size of an object module 433
sizeof 434
SMALL model . . . . Intel single-segment memory model 434
source file 435
sprintf() . . . . Print formatted text into a string 435
sqrt() . . . . Calculate the square root of a number 436
srand() . . . . Seed pseudo-random number generator 437
sscanf() . . . . Read and interpret text from a string 438
stack 440
Standard 440
standard error 440
standard input 441
standard output. 441
stat() . . . . Find file attributes 441
stat.h . . . . Definitions and declarations to obtain file status 443
statements 443
static . . . . Internal linkage 444
stdarg.h. . . . . Header for variable numbers of arguments 444
stderr . . . . Pointer to standard error stream 444
stdin . . . . Pointer to standard input stream. 445
STDIO. . . . . Standard input and output 445
stdio.h . . . . Declarations and definitions for STDIO 447
stdlib.h . . . . General utilities 447
stdout. . . . . Pointer to standard output stream 449
stime() . . . . Set the operating system time 449
storage-class specifiers. 450
storage duration. 450
strcat() . . . . Append one string onto another 451
strchr() . . . . Find a character in a string 451
strcmp(). . . . . Compare two strings 453
strcoll() . . . . Compare two strings, using locale-specific information. 453
strcpy() . . . . Copy one string into another 454
strcspn() . . . . Return length a string excludes characters in another 454
stream 455
strerror() . . . . Translate an error number into a string. 456
strftime() . . . . Format locale-specific time 457
string.h 458
string handling 459
string literal 460
strip . . . . Strip debug table from executable file 460
strlen() . . . . Measure the length of a string 461
strncat() . . . . Append *n* characters of one string onto another 461
strncmp(). . . . . Compare one string with a portion of another 462
strncpy() . . . . Copy one string into another 463

strpbrk() . . . . Find first occurrence of a character from another string 465
strrchr(). . . . . Search for rightmost occurrence of a character in a string. . . 466
strspn() . . . . Return length a string includes characters in another 467
strstr() . . . . Find one string within another 468
strtod() . . . . Convert string to floating-point number. 469
strtok() . . . . Break a string into tokens. 470
strtol(). . . . . Convert string to long integer 471
strtoul(). . . . . Convert string to unsigned long integer 472
struct 474
strxfrm() . . . . Transform a string 475
swab(). . . . . Swap a pair of bytes 475
switch. . . . . Select an entry in a table 476
system(). . . . . Suspend a program and execute another 477
tag. 478
tail. . . . . Print the end of a file 478
tan() . . . . Calculate tangent 478
tanh() . . . . Calculate hyperbolic tangent 479
technical information. 479
tempnam() . . . . Generate a unique name for a temporary file. 479
time() . . . . Get current calendar time 480
time . . . . Print current time/Time execution of a command. 480
time.h. . . . . Header for date and time 481
time_t . . . . Calendar time 481
time_to_jday() . . . . Convert system time to Julian date 482
TIMEZONE. . . . . Time zone information 482
tm . . . . Encode broken-down time. 484
tm_to_jday() . . . . Convert calendar format to Julian time 484
TMPDIR. . . . . Directory that holds temporary files 485
tmpfile(). . . . . Create a temporary file. 485
tmpnam(). . . . . Generate a unique name for a temporary file. 488
toascii() . . . . Convert characters to ASCII. 489
token 490
tolower() . . . . Convert character to lower case 491
toupper() . . . . Convert character to upper case 492
translation unit 493
trigraph sequences 493
true 494
typedef . . . . Synonym for another type. 494
type qualifier. 494
types 495
type specifier. 498
ungetc(). . . . . Push a character back into the input stream. 500
union 501
universal coordinated time 502
unlink(). . . . . Remove a file. 503
unsigned 504
unsigned char 504
unsigned int 504
unsigned long int 505
unsigned short int 505
va_arg() . . . . Return pointer to next argument in argument list. 506
va_end(). . . . . Tidy up after traversal of argument list 506
va_list. . . . . Type used to handle argument lists of variable length 507

va_start() . . . . Point to beginning of argument list 507
value preserving. 508
variable arguments 508
vfprintf() . . . . Print formatted text into stream 509
void . . . . Empty type. 511
void expression 513
volatile . . . . Qualify an identifier as frequently changing 513
vprintf() . . . . Print formatted text into standard output stream 513
vsprintf() . . . . Print formatted text into string 514
wc . . . . Count words, lines, and characters in files 516
wcstombs() . . . . Convert sequence of wide characters to multibyte characters . 516
wctomb() . . . . Convert a wide character to a multibyte character 517
while . . . . Loop construct 517
wildcards. 518
write() . . . . Write into a file 518
xctype.h 519
XOFF 519
XON. 519
xtime.h 519
Appendix 521