The MICROENGINE Company

CHANGE
NOTICE
79178

PRODUCT NAME: Pascal MICROENGINE Computer Users' Manual

DESCRIPTION OF CHANGE:

Page 10

The following should replace Table 2-4, Serial Port Baud Rate Switch Settings:

Baud Rate	Switch Settings Port A 1 2 3 4	Port B 5 6 7 8
50	0 0 0 0	0 0 0 0
75	1 0 0 0	0 0 0 1
110	0 1 0 0	0 0 1 0
134	1 1 0 0	0 0 1 1
150	0 0 1 0	0 1 0 0
200	1 0 1 0	0 1 0 1
300	0 1 1 0	0 1 1 0
600	1 1 1 0	0 1 1 1
1200	0 0 0 1	1 0 0 0
1800	1 0 0 1	1 0 0 1
2400	0 1 0 1	1 0 1 0
3600	1 1 0 1	1 0 1 1
4800	0 0 1 1	1 1 0 0
7200	1 0 1 1	1 1 0 1
9600	0 1 1 1	1 1 1 0
19200	1 1 1 1	1 1 1 1

Page 27

CHANGE NOTICE #79127, Page 3 discusses the Write/Not Read Signal.

IS:

W/R

CHANGED TO:

W/R-

Page 30

The following should be inserted in Section 5.2 after the description the four power supplies:

The power and ground connections to the PC board are:

A/B-1	A/B-2	A/B-3	A/B-4	A/B-5	A/B-6	A/B-7	A/B-8
GND	-5V	GND	+12V	-12V	GND	+5V	GND
A/B-50	A/B-49	A/B-48	A/B-47	A/B-46	A/B-45	A/B-44	A/B-43

Power Requirements:

+5v	2.5A
+12v	1.8A
-5v	12.0mA
-12v	.5A

Page 32

IS:

J1-15 or J2-15	DB	IXRC WD 1931

CHANGED TO:

J1-15 or J2-15	DB	IXTC WD 1931

Page 34

The following should replace Table 5-5, Serial Port Addresses (Please note this supersedes CHANGE NOTICE #79127):

Operation	General Address	Value of "x"	Element Selected
Input from Serial Port A	FC1x	0 - 0 1	Status Register
		0 - 0 0	Receiver Holding Register
Output to Serial Port A	FC1x	0 - 0 0	Transmitter Holding Register
Activate System Terminal Specification Switch	FC1x	0 - - -	-
Input from Serial Port B	FC2x	- - 0 1	Status Register
		- - 0 0	Receiver Holding Register
Output to Serial Port B	FC2x	- - 0 0	Transmitter Holding Register

Page 42

The following should replace Table 5-9, Serial Port SYN, DLE, and Control Register Addresses:

Operation	General Address	Value of "x"	Element Selected
Input from Serial Port A	FC1x	0 - 1 1	Control Register 1
		0 - 1 0	Control Register 2
Output to Serial Port A	FC1x	0 - 1 1	Control Register 1
		0 - 1 0	Control Register 2
		0 - 0 1	SYN and DLE Registers
Input from Serial Port B	FC2x	- - 1 1	Control Register 1
		- - 1 0	Control Register 2
Output to Serial Port B	FC2x	- - 1 1	Control Register 1
		- - 1 0	Control Register 2
		- - 0 1	SYN and DLE Registers

Page 44

IS:

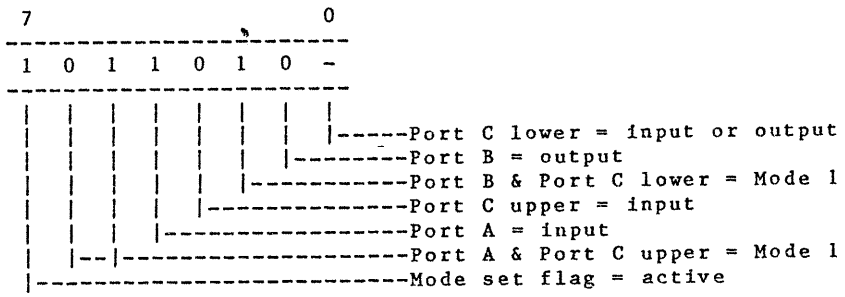
J3-6	SACKB-	ACKNOWLEDGE B

CHANGED TO:

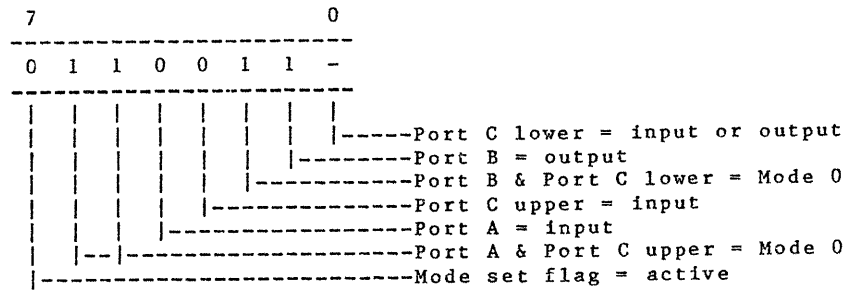
J3-6	SACKB-	SELECT ACKNOWLEDGE B POLARITY

Page 48

IS:



CHANGED TO:



Page 50

IS:

J4-3	DS3	UNIT SELECT 2

CHANGED TO:

J4-3	DS3	UNIT SELECT 3

The MICROENGINE Company

CHANGE NOTICE 79178

PRODUCT NAME: Pascal MICROENGINE Computer Pascal Operations Manual

DESCRIPTION OF CHANGE:

OUTLINE

Under Appendix B, delete B.6 Assembler Syntax Errors

Page 8

IS:

The cursor moves to the start of the top line.

CHANGED TO:

The cursor moves to the same relative page position as it was previously, or to the start or the end of the file.

Page 17

Delete the following sentence:

Marker names are case-sensitive; upper and lower cases of the same letter are considered to be different letters.

IS:

If a copy or deletion is made between the beginning of the file and the position of the marker, the absolute position of the marker will be changed.

CHANGED TO:

If an insertion, copy, deletion, or zap is made between the beginning of the file and the position of the marker, the marke is adjusted so that a JUMP to it is to the same place in the text. If the text which includes a marker is deleted, a subsequent JUMP to the marker is to the start of the text following the deletion, if any, or to the end of the file.

ge 32

IS:

The rules for legal file and volume names are given in Section 3.1, Files.

CHANGED TO:

The rules for legal file and volume names are given in Section 3.2, Files.

ge 34

IS:

The SAVE command is specified by typing "S" for S(ave. SAVE...

CHANGED TO:

The SAVE command is specified by typing "S" for S(ave. If the workfile was established by GET <file name> the File Handler prompts:

Save as <file name>?

If the response is "Y", the work file is saved under <file name>. If the response is not "Y", or the work file was not established by the GET command, the File Handler...

IS:

For example, if the operator enters "BLACK:BART" in response to the prompt "Save as what file?", the File Handler will generate the prompt line:

Would you like BART.TEXT written to BLACK:?

A "Y" answer to this prompt will cause the File Handler to attempt a transfer of the work file to the specified volume and file (see TRANSFER).

CHANGED TO:

If the destination file is not the system disk, the save of the work file is the equivalent of a transfer of the work file to the destination file (see TRANSFER).

Page 36

IS:

HIGH:START.TEXT changed to LOWRT.END
HIGH:STOPS.TEXT changed to LOWPS.END

CHANGED TO:

HIGH:START.TEXT changed to LOWART.END
HIGH:STOPS.TEXT changed to LOWOPS.END

Page 43

The following should replace the description of the ZERO command given on pages 42 and 43:

The ZERO command is specified by typing "Z" for Z(ero. ZERO initializes the directory on the specified volume with a new volume ID and with all blocks on the disk unused. When "Z" is typed, the File Handler responds with the following prompt line:

Zero dir of what vol?

The user then enters the current volume ID. If the disk has not been previously ZEROed, the second prompt is:

Duplicate dir?

If the answer is "Y", a duplicate directory will be maintained. The primary directory is in blocks 2 through 5, and if maintained, the duplicate directory is in blocks 6 through 9. In case the primary directory is destroyed, the disk can be restored from the duplicate directory using the utility COPYDUPDIR. The next prompt is:

# of blocks?

The number of blocks on the disk is entered according to Table 2-1, which gives the size in blocks for various types of disks. The File Handler then asks:

New vol name?

and queries to verify the given name:

<new volume name> correct?

If "Y" is given, the disk is ZEROed; otherwise, it is not altered from its original state. In either case, control returns to the outer level of the File Handler.

If the specified disk has been previously ZEROed, the following changes apply to the prompt sequence. Before prompting for duplicate directory, the File Handler queries:

Destroy <current volume name>?

If the answer is "N" control returns to the outer level of the File Handler. Instead of asking for the number of blocks, the File Handler prompts:

<current number of> blocks?

If the answer is "Y" the ZERO sequence continues to the new volume name; otherwise, the number of blocks is first determined the same as for a non-ZEROed disk.

Page 49

IS:

```
USES UNITA, UNITD, Found in *SYSTEM.LIBRARY
  $U NEW.CODE
     UNITB
  $U OLD.CODE
     UNITC,UNITE;
```

CHANGED TO:

```
USES UNITA, UNITD, {Found in *SYSTEM.LIBRARY}
  {$U NEW.CODE}
     UNITB,
  {$U OLD.CODE}
     UNITC,UNITE;
```

Page 54

IS:

Lib file? <codefile identifier>

CHANGED TO:

Lib file? <codefile identifier><ret>

Page 55

IS:

```
RECORD
  DISKINFO: ARRAY[0 . . 15] OF
              RECORD
                CODELENG, CODEADDR: INTEGER
              END
  SEGNAME: ARRAY[0 . . 15] OF PACKED ARRAY[0 . . 7] OF CHAR;
  SEGKIND: ARRAY[0 . . 15] OF (LINKED,HOSTSEG,SEGPROC,UNITSEG,
                               SEPRTSEG);
  TEXTADR: ARRAY[0 . . 15] OF INTEGER;
END
```

CHANGED TO:

```
RECORD
  DISKINFO: ARRAY[0..15] OF
              RECORD
                CODELENG,CODEADDR:INTEGER
              END;
  SEGNAME:  ARRAY[0..15] OF PACKED ARRAY[0..7] OF CHAR;
  SEGKIND:  ARRAY[0..15] OF (LINKED,HOSTSEG,SEGPROC,UNITSEG,
                             SEPRTSEG);
  END;
```

Page 56

IS:

SEPTRSEG A separately compiled procedure of function (e.g., assembly language code files or Pascal UNITS that are not SEGMENT UNITs.)

CHANGED TO:

SEPTRSEG A separately compiled procedure or function (e.g., Pascal UNITS that are not SEGMENT UNITS.)

IS:

```
    END(lientry)
```

CHANGED TO:

```
    END(*lientry*)
```

Page 58

IS:

After "D" is typed, the Debugger displays a message giving the release number and the date of the release:

PASCAL INTERACTIVE DEBUGGER - January 1978

CHANGED TO:

After "D" is typed, the Debugger announces itself and displays release number.

Page 64

IS:

UNITNUMBER :physical device number used to determine device handler used by the interpreter

CHANGED TO:

UNITNUMBER :physical device number

Page 68

IS:

RESET (FILEID) without an optional string parameter rewinds the file by setting the file pointers back to the beginning (0 record) of the file. The boolean functions EOF and EOLN will not be set by the implied GET in RESET.

With files of the INTERACTIVE type, these functions act differently. On files of other types, RESET will do an initial GET to the file, setting the window variable to the first record in the file.

CHANGED TO:

RESET (FILEID) without an optional string parameter rewinds the file by setting the file pointers back to the beginning (0 record) of the file. The boolean functions EOF and EOLN will now be set by the implied GET in RESET.

With files of the type INTERACTIVE, RESET acts differently. On files of other types RESET will do an initial GET on the file, setting the window variable to the first record in the file.

IS:

PROCEDURE SEEK(FILEID,INTEGER);

SEEK changes the file pointers so that the next GET or PUT uses the INTERGER the record of FILEID.

CHANGED TO:

PROCEDURE SEEK(FILEID,INTEGER);

SEEK changes the file pointers so that the next GET or PUT use the INTEGERth record of FILEID.

IS:

FUNCTION UNITBUSY(UNITNUMBER): BOOLEAN

CHANGED TO:

FUNCTION UNITBUSY(UNITNUMBER): BOOLEAN;

Page 69

IS:

If BLOCK- NUMBER is omitted, but INTEGER is included, a comma used to hold the placement of parameters.

CHANGED TO:

If BLOCKNUMBER is omitted, but INTEGER is included, a comma is used to hold the placement of parameters.

The following should be inserted at the end of Section 3.1.2 (after UNITWAIT):

FUNCTION BLOCKREAD(FILEID,ARRAY,BLOCKS,[RELBLOCK]): INTEGER;
FUNCTION BLOCKWRITE(FILEID,ARRAY,BLOCKS,[RELBLOCK]): INTEGER;

These functions return an INTEGER value of the number of blocks of data transferred. The FILE must be an untyped file. The length of ARRAY should be an integer multiple of bytes-per-disk block. BLOCK$ is the number of blocks to be transferred. RELBLOCK is the blocknumber relative to the start of the file, block zero being the first block. If no RELBLOCK is specified, the I/O will be done sequentially. A random access I/O moves the file pointers. EOF(FILEID) becomes true when the last block in the file is read.

Page 71

IS:

FUNCTION POS(STRING,SOURCE): INTEGER

CHANGED TO:

FUNCTION POS(STRING,SOURCE): INTEGER;

IS:

PROCEDURE DELETE(DESTINATION,INDEX,SIZE): STRING

CHANGED TO:

PROCEDURE DELETE(DESTINATION,INDEX,SIZE);

The following should be inserted at the end of Section 3.1.3 (after INSERT):

PROCEDURE STR(LONG,DESTINATION);

This procedure converts a long integer LONG into a string. The resulting string is placed in DESTINATION. The integer LONG may also be a normal INTEGER.

Page 73

IS:

The first word of the pair gives the blocknumber within the segment where code begins.

CHANGED TO:

The first word of the pair gives the blocknumber within the file where the code begins.

Page 76

CHANGE NOTICE #79127, Page 3, mentions Figure 3-3 as an example of Pascal to Pascal linkage. Delete the following two sentences as the example is not given:

An example of Pascal to Pascal linkage is shown in Figure 3-3. As can be seen, the program must indicate the UNITS it USES before the LABEL decaration part of the program.

Page 77

Delete the following two sentences as Figure 3-5 is not an example of UNIT using another UNIT:

A UNIT may use another UNIT, as shown in the example in Figure 3-5. In this case, the USES declarartion must appear at the beginning of the INTERFACE part.

Page 81

IS:

The program in Figure 3-9 illustrates how MARK and RELEASE can

be used to change the size of the heap.

CHANGED TO:

The program in Figure 3-7 illustrates how MARK and RELEASE can be used to change the size of the heap.

Page 87

IS:

A list of file parameters may follow the file identifier.

CHANGED TO:

A list of file parameters may follow the program identifier.

Page 88

IS:

The standard definition of the procedure READ requires tha the process of opening a file include loading the window variable F^ with the first character of the file.

CHANGED TO:

The standard definition of the procedure READ requires that the process of opening a file include loading the window variable F^ with the first character of the file.

IS:

For example, the following two statements are equivalent to READ(INPUT,CH

CHANGED TO:

For example, the following two statements are equivalent to READ(INPUT,CH);

Page 89

IS:

This value can be overridden in the declaration of a string by appending the desired length within [] after the type iden-tifier.

CHANGE TO:

This value can be overridden in the declaration of a string by appending the desired length within [] after the type identifier.

Page 90

IS:

For writing variables of type STRING, see Section 3.1.1, String Intrinsics.

CHANGED TO:

For writing variables of type STRING, see Section 3.1.3, String Intrinsics.

Page 93

IS:

HAS CLOCK

If TRUE, a real-time clock is available; otherwise FALSE. The real-time clock module is assumed to be a line frequency clock. When available, the clock is used by the system to optimize disk directory updates (see Section 3.1.6, Time Intrinsics).

CHANGED TO:

HAS CLOCK

Should be set FALSE as the Pascal MICROENGINE has no real-time clock.

Page 94

IS:

PREFIXED[KKEY FOR MOVING CURSOR UP] TRUE

CHANGED TO:

PREFIXED[KEY FOR MOVING CURSOR UP] TRUE

Page 97

The following should replace Section 4.2 (BOOTSTRAP COPIER):

4.2 BOOTSTRAP UTILITIES

There are two methods for placing a bootstrap on a disk. The BOOTER utility will copy a bootstrap from an existing disk to a specified unit. The BOOTMAKE utility will create a bootstrap. When making a complete copy of a bootable disk it is necessary to use one of these utilities to place a bootstrap on the destination disk. The T(ransfer command in

the Filer will not completely copy the bootstrap as track 0 is not normally accessed by UCSD software but is reserved for the bootstrap.

4.2.1 BOOTER

This utility will copy a bootstrap from an existing unit to a specified unit. The BOOTER.CODE utility is run by typing an "X" for eX(ecute at the Outer Level of commands, followed by BOOTER. The following prompt will appear:

```
Unit to write boot to [4,5,9,10]:
```

Enter the number for the destination disk and a <ret>. Then this prompt will be displayed:

```
Unit boot is on [4,5,9,10]:
```

Enter the number for the disk that has the existing bootstrap and a <ret>. BOOTER will then indicate that the bootstrap was transferred.

4.2.2 BOOTMAKE

BOOTMAKE makes bootable programs of two types, diagnostic and system. A diagnostic boot runs stand-alone (without an operating system). A diagnostic boot resides on all of track 1 (blocks 0-6). A system boot is one that loads and executes the operating system. It resides on blocks 0 and 1 and all of track 0. In order to run BOOTMAKE.CODE, type an "X" for eX(ecute in the Outer Level of commands, followed by BOOTMAKE. BOOTMAKE then prompts with the following sequence:

```
Enter code file name
Want a hex listing of files (Y or N)
D(iagnostic or S(ystem Boot?
Enter unit number where boot to be placed
```

The code file name is the code file with the diagnostic or system bootstrap. The system bootstrap is BOOT.CODE. If a listing of files is selected a printer must be on line. This option then dumps the core image of the bootstrap in hexadecimal to the line printer. The next prompt selects between a diagnostic or a system bootstrap. Note that a diagnostic bootstrap will over-write the disk directory of the unit where the bootstrap is placed. The final option selects the unit on which to place the bootstrap.

IS:

If the disk is not currently maintaining a current directory, a message is generated.

CHANGED TO:

If the disk is not currently maintaining a duplicate directory, a message is generated.

Delete the following two sentences as the RECOVERY program is not currently available:

If the disk is not currently maintaining a current directory, the RECOVER program should be run. This is described

Page 98

IS:

The program will check for this and will give the following message if the blocks appear in use.

CHANGED TO:

The program will check for this and will generate a message should the blocks appear in use.

Page 99

IS:

The library map utility usually is used to list library definitions. However, when the program prompts for a reference list.

CHANGED TO:

The library map utility usually is used to list library definitions. However, when the program prompts for a reference list an option is available to include intra-library symbol references. Should this feature be desired, type a "Y" when queried for a reference list.

Page 104

IS:

THE CALCULATOR

The calculator program is entered by typing "X" for eX(ecute while in the Outer Level of commands.

CHANGED TO:

THE CALCULATOR

The calculator program is entered by typing "X" for eX(ecute while in the Outer Level of commands, followed by CALC.

Page 106

IS:

To create "local GOTOXY", examine the file GOTOXY.TEXT, that is on the release disk, with the Screen-Oriented Editor. This file contains a few procedures for doing GOTOXY cursor addressing on several types of video terminals. If the procedure needed is in the file, remove it from comments, comment out any other procedures, recompile it and run BINDER on it. Directions for entering it are given above. Possible errors that may occur while reviewing the GOTOXY.TEXT and selecting the procedure needed are:

Possible Error	Fix
Nil memory reference at compile time	Remove the program heading and try again
Value range error when executing BINDER	(**U-*) should be the first thing in the GOTOXY file.

If the needed procedure is not in the file, it must be created. The created procedure can not be named GOTOXY because this identifier is predeclared at the "$U-" level of compilation.

CHANGED TO:

To create "local GOTOXY", examine the example GOTOXY procedure below. It is the procedure for doing GOTOXY cursor addressing for the Soroc IQ120 terminal:

```
PROGRAM EXAMPLE ;
{GOTOXY FOR SOROC IQ120}
PROCEDURE IQ120XY(X,Y: INTEGER);
VAR P: PACKED ARRAY[0..3] OF CHAR;
BEGIN
  IF Y>23 THEN Y:=23;
  IF X>79 THEN X:=79;
  IF Y<0 THEN Y:=0;
  IF X<0 THEN X:=0;
  P[0]:=CHR(27);
  P[1]:='=';
  P[2]:=CHR(Y+32);
  P[3]:=CHR(X+32);
  UNITWRITE(2,P,0);
END;
BEGIN END.
```

Modify this to meet the specifications of the intended terminal, recompile it and run BINDER on it. GOTOXY must be the only procedure declared within the source program (other than the dummy system main program which should be empty). The GOTOXY

procedure must not make use of any STRING or REAL constants.
It is necessary to run SETUP on the newly produced system
codefile in order for GOTOXY to work properly.

Page 132

The syntax diagram for FACTOR is in error:

In line 3, the square brackets [] around expression should be
parenthesis ().
In line 6, the square brackets [] around expression should be
parenthesis ().
In line 8, the aquare brackets [] surrounding the last part of
the diagram should be parenthesis ().

The syntax diagram for PARAMETER LIST is in error:

In line 3, the square brackets [] surrounding both identifier
and "type" identifier should be parenthesis ().