

**Library: emu8086.inc**

To use any of the functions in emu8086.inc you should have the following line in the beginning of your source file:

```
include 'emu8086.inc'
```

emu8086.inc defines the following **macros**:

PUTC char	macro with 1 parameter, prints out an ASCII char at current cursor position.
GOTOXY col, row	macro with 2 parameters, sets cursor position.
PRINT string	macro with 1 parameter, prints out a string.
PRINTN string	macro with 1 parameter, prints out a string. The same as PRINT but automatically adds "carriage return" at the end of the string.

```
05 PRINT "HELLO"    ;prints the string
06 PUTC 'W'         ;prints 'W' next to 'O'
07
08 GOTOXY 20, 3     ;go to the given position
09
10 PRINTN "HELLO"   ;prints the string at the cursor position (20, 3) and
11                  ;prints a carriage return and newline
12
13 PUTC 65           ;prints the char whose ascii is 65
14
15
```

emu8086.inc also defines the following procedures:

- **PRINT_STRING** - procedure to print a null terminated string at current cursor position, receives address of **string in DS:SI register**. To use it declare: **DEFINE_PRINT_STRING** before END directive.
- **PTTHIS** - procedure to print a null terminated string at current cursor position (just as PRINT_STRING), but receives address of **string from Stack**. The NULL TERMINATED string **should be defined just after the CALL instruction**. For example:

```
CALL PTHIS  
db 'Hello World!', 0
```

To use it declare: **DEFINE_PTHIS** before END directive.

- **GET_STRING** - procedure to get a null terminated string from the user, the received string is written to buffer at **DS:DI**, buffer **size should be in DX**. Procedure **stops the input when 'Enter' is pressed**. To use it declare: **DEFINE_GET_STRING** before END directive.
- **CLEAR_SCREEN** - procedure to clear the screen, (done by scrolling the entire screen window), and set cursor position to top of it. To use it declare: **DEFINE_CLEAR_SCREEN** before END directive.
- **SCAN_NUM** - procedure that gets the multi-digit SIGNED number from the keyboard, and **stores the result in CX register**. To use it declare: **DEFINE_SCAN_NUM** before END directive.
- **PRINT_NUM** - procedure that prints **a signed number in AX register**. To use it declare: **DEFINE_PRINT_NUM** and **DEFINE_PRINT_NUM_UN** before END directive.
- **PRINT_NUM_UN** - procedure that prints out an unsigned number **in AX register**. To use it declare: **DEFINE_PRINT_NUM_UN** before END directive.

To use any of the above procedures you should first declare the function in the bottom of your file (but before the **END** directive), and then use **CALL** instruction followed by a procedure name. For example:

```

02 INCLUDE 'EMU8086.INC'
03 .MODEL SMALL
04 .STACK 100H
05 .DATA
06     MSG DB "This is 'print_string' procedure", 0
07
08
09 .CODE
10     MOV AX, @DATA
11     MOV DS, AX
12
13     LEA SI, MSG           ;load offset of the string to SI to print
14     CALL PRINT_STRING
15
16     CALL PTHIS           ;print the string declared in code segment
17     DB 0AH, 0DH, "This is 'pthis' procedure", 0AH, 0DH, 0
18
19     MOV DX, 5             ;place the input buffer size
20     CALL GET_STRING      ;input string
21
22     CALL CLEAR_SCREEN    ;clear the screen and set cursor position at (0,0)
23
24     CALL SCAN_NUM        ;get any decimal number (can be multidigit), ends with an enter
25
26     CALL PTHIS           ;print a newline and caret
27     DB 0AH, 0DH, 0
28
29     MOV AX, CX            ;move the input number from cx to ax to print
30     CALL PRINT_NUM_UNNS  ;unsigned number print
31
32     EXIT:
33     MOV AH, 4CH
34     INT 21H
35
36 ;-----DEFINE LIBRARY PROCUDERS-----;
37
38     DEFINE_PRINT_STRING
39     DEFINE_PTHIS
40     DEFINE_GET_STRING
41     DEFINE_CLEAR_SCREEN
42     DEFINE_SCAN_NUM
43     DEFINE_PRINT_NUM
44     DEFINE_PRINT_NUM_UNNS
45
46 END

```

Example: take two numbers as input and display their summation.

```
01 INCLUDE 'EMU8086.INC'
02 .MODEL SMALL
03 .STACK 100H
04 .DATA
05     MSG1 DB 'Enter a number: ', 0
06     MSG2 DB 0AH, 0DH, 'Enter another number: ', 0
07 .CODE
08     MOV AX, @DATA
09     MOV DS, AX
10
11     LEA SI, MSG1           ; ask for the number
12     CALL PRINT_STRING
13     CALL SCAN_NUM         ; get 1st number in CX.
14
15     MOV AX, CX            ; copy the number to AX.
16
17     LEA SI, MSG2           ; ask for the number
18     CALL PRINT_STRING
19     CALL SCAN_NUM         ; input 2nd number in CX
20
21     ADD AX, CX            ; sum in AX
22
23     CALL PTHIS
24     DB 13, 10, 'The sum is: ', 0
25
26     CALL PRINT_NUM        ; print number in AX
27
28     EXIT:                 ; return to operating system.
29     MOV AH, 4CH
30     INT 21H
31 ;-----define-----;
32 DEFINE_SCAN_NUM
33 DEFINE_PRINT_STRING
34 DEFINE_PRINT_NUM
35 DEFINE_PRINT_NUM_UN$    ; required for print_num.
36 DEFINE_PTHIS
37 END                     ; directive to stop the compiler.
```

Example: Read a string using GET_STRING and store it in memory.

```
01
02 INCLUDE 'EMU8086.INC'
03
04 .MODEL SMALL
05 .STACK 100H
06 .DATA
07     MSG DB "This is 'print_string' procedure", 0
08     STR DB 5 DUP(?)
09
10 .CODE
11     MOV AX, @DATA
12     MOV DS, AX
13
14     MOV DX, 5                ;declare the buffer size
15     CALL GET_STRING          ;input string
16     LEA BX, STR              ;load the offset of memory into index reg.
17
18     COPY:                    ;copy the input from buffer to memory
19     MOV AL, [DI]             ;copy each char of the string individually
20     MOV [BX], AL
21
22     INC BX                   ;update memory offset
23     INC DI                   ;update DI to get the next user input char
24
25     CMP [DI], 0              ;check if it is the end of user input
26     JNE COPY
27
28     MOV [BX], 0              ;place null character at the end of the string
29
30     CALL PTHIS               ;print newl
31     DB 0AH, 0DH, 0
32
33     LEA SI, STR              ;load offset of the string to SI to print
34     CALL PRINT_STRING
35
36     EXIT:
37     MOV AH, 4CH
38     INT 21H
39
40 ;-----DEFINE LIBRARY PROCUDERS-----;
41
42     DEFINE_PRINT_STRING
43     DEFINE_PTHIS
44     DEFINE_GET_STRING
45
46 END
47
```