

**debug\_lib**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> debug_lib		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 29, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>debug_lib</b>	<b>1</b>
1.1	debug_lib.doc . . . . .	1
1.2	debug.lib/KCmpStr . . . . .	1
1.3	debug.lib/KGetChar . . . . .	1
1.4	debug.lib/KGetNum . . . . .	2
1.5	debug.lib/KMayGetChar . . . . .	2
1.6	debug.lib/KPrintf . . . . .	2
1.7	debug.lib/KPutChar . . . . .	3
1.8	debug.lib/KPutStr . . . . .	3

# Chapter 1

## debug\_lib

### 1.1 debug\_lib.doc

```
KCmpStr()  
KGetChar()  
KGetNum()  
KMayGetChar()  
KPrintf()  
KPutChar()  
KPutStr()
```

### 1.2 debug.lib/KCmpStr

NAME

KCmpStr - compare two null terminated strings

SYNOPSIS

```
mismatch = KCmpStr(string1, string2)
```

```
D0          A0          A1
```

FUNCTION

string1 is compared to string2 using the ASCII collating sequence. 0 indicates the strings are identical.

### 1.3 debug.lib/KGetChar

NAME

KGetChar - get a character from the console  
(defaults to the serial port at 9600 baud)

SYNOPSIS

```
char = KGetChar()
```

```
D0
```

FUNCTION

busy wait until a character arrives from the console.

---

KGetChar is the assembly interface, `_KGetChar` and `_kgetc` are the C interfaces.

## 1.4 debug.lib/KGetNum

### NAME

KGetNum - get a number from the console

### SYNOPSIS

```
number = KGetNum()  
D0
```

### FUNCTION

get a signed decimal integer from the console. This will busy wait until the number arrives.

## 1.5 debug.lib/KMayGetChar

### NAME

KMayGetChar - return a character if present, but don't wait  
(defaults to the serial port at 9600 baud)

### SYNOPSIS

```
flagChar = KMayGetChar()  
D0
```

### FUNCTION

return either a -1, saying that there is no character present, or whatever character was waiting. KMayGetChar is the assembly interface, `_KMayGetChar` is the C interface.

## 1.6 debug.lib/KPrintf

### NAME

KPrintf - print formatted data to the console  
(defaults to the serial port at 9600 baud)

### SYNOPSIS

```
KPrintf("format string", values)  
      A0                A1
```

### FUNCTION

print a formatted C-type string to the console. See the `exec RawDoFmt()` call for the supported % formatting commands.

### INPUTS

"format string" - A C style string with % commands to indicate where paramters are to be inserted.  
values - A pointer to an array of paramters, to be inserted into specified places in the string.

---

KPutFmt and KPrintf are identical assembly interfaces that want the two pointers in registers. \_KPrintf and \_kprintf are C interfaces that expect the format string pointer on the stack, and the parameters on the stack above that.

SEE ALSO  
exec.library/RawDoFmt, any C compiler's "printf" call.

## 1.7 debug.lib/KPutChar

NAME  
KPutChar - put a character to the console  
(defaults to the serial port at 9600 baud)

SYNOPSIS  
char = KPutChar(char)  
D0                    D0

FUNCTION  
Put a character to the console. This function will not return until the character has been completely transmitted.

INPUTS  
KPutChar is the assembly interface, the character must be in D0. \_KPutChar and \_kputc are the C interfaces, the character must be a longword on the stack.

## 1.8 debug.lib/KPutStr

NAME  
KPutStr - put a string to the console  
(defaults to the serial port at 9600 baud)

SYNOPSIS  
KPutStr(string)  
A0

FUNCTION  
put a null terminated string to the console. This function will not return until the string has been completely transmitted.

INPUTS  
KPutStr is the assembly interface, a string pointer must be in A0. \_KPutStr and \_kputs are the C interfaces, the string pointer must be on the stack.

---