

Operation, Page 7

The console serial port configuration should read as follows:

The port is configured for 9600 baud, 8 data bits, 1 stop bit, no parity. Neither software flow control or hardware handshaking are implemented on this port.

RS-485 Configuration, Page 13

The COM2 port of the Flashlite can be configured and used for RS-485 communications. In order to avoid conflicts with DOS and the BIOS, it is first necessary to move the console to COM1. This is done using the utility program CON2COM1. Please note that COM1 (J6) is pinned out as DTE and you must use a null modem cable to connect it to a PC serial port.

To enable RS-485 operation (and disable RS-232) on COM2, clear bit 6 of I/O port F872 hex.

```
#define EN485_MASK 0xBF
#define EN485_REG 0xF872
outportb(EN485_REG, (inportb(EN485_REG) & EN485_MASK) ); // change to RS-485
```

Bit 0 of the PINCFG register must be set to allow control of the RS-485 transmit enable pin. The PINCFG register is located at I/O port F826 hex.

```
#define PINCFG 0xF826
outportb(PINCFG, (inportb(PINCFG) | 0x01) ); // connect TE control to chip pkg
```

To transmit RS-485 data, set bit 1 of I/O port F8FC hex. To receive RS-485 data, clear bit 1 of I/O port F8FC hex.

```
#define TX_MASK 0x02
#define TX_MASK_REG 0xF8FC
outportb(TX_MASK_REG, (inportb(TX_MASK_REG) | TX_MASK) ); // enable transmitter
outportb(TX_MASK_REG, (inportb(TX_MASK_REG) & ~TX_MASK) ); // disable transmitter
```

In all cases, be sure to preserve the other bit settings of these ports. Bits are numbered 7-0, bit 7 is the most significant bit.

QuickBASIC Console I/O, Page 22

Some of the code produced by Microsoft QuickBASIC and QuickBASIC Professional compilers does not execute properly on the Flashlite. In the case of console I/O, we believe that QuickBASIC is generating code for specific hardware and software not present on the Flashlite controller.

There are two problems with console I/O. The first is that a PRINT statement will not send output to the console port. To output text to the console, open "cons:" as a file and print to it.

```
OPEN "o", 1, "cons:"
PRINT #1, "This is a test of Console output"
```

The second problem is that an INPUT statement will not echo data entered by the user. To work around this problem, use the line input function listed below.

```
rem linein returns the echoed string of text linein$ from the console. If you need the text
rem in the form of a number, use VAL(linein$) to convert the string to a number
```

```
linein:
linein$ = ""
linemore:
a$ = INKEY$
IF a$ = "" THEN GOTO linemore
IF a$ = CHR$(13) THEN GOTO linedone
IF a$ <> CHR$(8) THEN GOTO getchar
PRINT #1, CHR$(8); CHR$(32); CHR$(8);
linein$=left$(linein$, (len(linein$)-1))
GOTO linemore
getchar:
PRINT #1, a$;
linein$ = linein$ + a$
GOTO linemore
linedone:
PRINT #1, ""
RETURN
```