

## 15.17 Tty Output Interrupt Processing (`ttyhandle_out`)

Output interrupt processing is the easiest to understand. When an output interrupt occurs, the device has transmitted all characters from the onboard FIFO and is ready for more. *Ttyhandler* clears the interrupt and calls `ttyhandle_out` to restart output. The code for `ttyhandle_out` can be found in file `ttyhandle_out.c`.

```

/* ttyhandle_out.c - ttyhandle_out */

#include <xinu.h>

/*-----
 * ttyhandle_out - Handle an output on a tty device by sending more
 *                  characters to the device FIFO (interrupts disabled)
 *-----
 */
void ttyhandle_out(
    struct ttyblk *typtr,          /* Ptr to ttytab entry */
    struct uart_csreg *csrptr     /* Address of UART's CSRs */
)
{

    int32  ochars;                 /* Number of output chars sent */
                                     /* to the UART */
    int32  avail;                 /* Available chars in output buf*/
    int32  uspace;                /* Space left in onboard UART */
                                     /* output FIFO */
    byte   ier = 0;

    /* If output is currently held, simply ignore the call */

    if (typtr->tyoheld) {
        return;
    }

    /* If echo and output queues empty, turn off interrupts */

    if ( (typtr->tyehead == typtr->tyetail) &&
          (semcount(typtr->tyosem) >= TY_OBUFLLEN) ) {
        ier = csrptr->ier;
        csrptr->ier = ier & ~UART_IER_ETBEI;
        return;
    }

    /* Initialize uspace to the size of the transmit FIFO */

```