









**Listing 1**

```
void Task1 (void *data)
{
  UBYTE err;
  INT16U os_version;
  INT8U nbytes;
  INT8U c;
  char s[81];
  char *ps;
  int result;

  DispInit(2,40); /* Init LCD Display 2x40 */
  DispClrScr();

  CommInit(); /* Init COMM port */
  CommCfgPort(COMM1, 9600, 8, COMM_PARITY_NONE, 1);
  CommSetIntVect(COMM1);
  CommRxIntEn(COMM1);
  data = data; /* Prevent compiler warning */
  for (;;) {
    union REGS regs; /* TS-2800 Bios Call */
    ps = s;
    nbytes = 0;
    do {
      c = CommGetChar(COMM1, OS_TICKS_PER_SEC, &err);
      *ps++ = c;
      nbytes++;
    } while (c != '\n' && nbytes < 1);
    *ps = NUL; /* NUL terminate received string */
    OSSemPend(LCDSem, 0, &err); /* Get Semaphore */
    DispStr(0, 0, "cmd:"); /* Print CMD's to LCD Display */
    DispStr(0, 5, s);
    OSSemPost(LCDSem); /* Release Semaphore */

    switch (c) {
      case 'a': /* Turn LED ON. */
      {
        regs.x.ax = 0xB010;
        regs.h.bh = 0x01;
        int86 (0x15, &regs, &regs);
        printf ("Led = On \n");
      }
      break;
      case 'b': /* Turn LED OFF.*/
      {
        regs.x.ax = 0xB010;
        regs.h.bh = 0x00;
        int86 (0x15, &regs, &regs);
        printf ("Led = Off \n");
      }
      break;
      case 'c': /* Set P1.0 = Digital Output 1 */
      {
        result = (inpw(0xf820)) & 0x0fe;
        outpw(0xf820, result);
        result = (inpw(0xf864)) & 0x0fe;
        outp(0xf864, result);
        result = (inpw(0xf862)) | 0x01;
        outpw(0xf862, result);
        printf ("Digital out 1 = Set \n");
      }
      break;
      case 'd': /* Clear P1.0 = Digital Output 1 */
      {
        result = (inpw(0xf862)) & 0xfe;
        outpw(0xf862, result);
        printf ("Digital out 1 = Clear \n");
      }
      break;
      case 'e': /* Set P1.5 = Digital Output 2 */
      {
        result = (inpw(0xf820)) & 0x0df;
        outpw(0xf820, result);
        result = (inpw(0xf864)) & 0x0df;
        outp(0xf864, result);
        result = (inpw(0xf862)) | 0x20;
      }
    }
  }
}
```

Continue

Circuit Cellar, the Magazine for Computer Applications.  
Reprinted by permission. For subscription information,  
call (860) 875-2199, subscribe@circuitcellar.com or  
www.circuitcellar.com/subscribe.htm.

**Listing 1– Continued**

```
outpw(0xf862, result);
    printf (“Digital out 2 = Set \n”);
    break; }
    case ‘f’: /* Clear P1.5 = Digital Output 2 */
    {
        result = (inpw(0xf862)) & 0xdf;
outpw(0xf862, result);
    printf (“Digital out 2 = Clear \n”);
    break; }
    case ‘g’: /* Set P3.2 = Digital Output 3 */
    {
        result = (inpw(0xf824)) & 0x0fb;
outpw(0xf824, result);
        result = (inpw(0xf874)) & 0x0fb;
        outp(0xf874, result);
        result = (inpw(0xf872)) | 0x04;
        outpw(0xf872, result);
        printf (“Digital out 3 = Set \n”);
    break; }
    case ‘h’: /* Clear P3.2 = Digital Output 3 */
    {
        result = (inpw(0xf872)) & 0x0fb;
        outpw(0xf872, result);
        printf (“Digital out 3 = Clear \n”);
    break; }

    case ‘i’: /* Set P3.6 = Digital Output 4 */
    {
        result = (inpw(0xf824)) & 0x0bf;
outpw(0xf824, result);
        result = (inpw(0xf874)) & 0x0bf;
        outp(0xf874, result);
        result = (inpw(0xf872)) | 0x40;
        outpw(0xf872, result);
        printf (“Digital out 4 = Set \n”);
    break; }
    case ‘j’: /* Clear P3.6 = Digital Output 4 */
    {
        result = (inpw(0xf872)) & 0x0bf;
outpw(0xf872, result);
        printf (“Digital out 4 = Clear \n”);
        break; }
    case ‘k’: /* Display OS version */
    {
        os_version = OSVersion();
        printf (“uCOS-II ver = %d\n”, os_version);
        break; }
    }
}
```

**Listing 2**

```
void Task2 (void *data)
{
    UBYTE err;
    int msb;
    int lsb;
    int DigitalInput1;
    char s[80];
    char *ps;
    data = data;
    for (;;) {
```

Continued

**Listing 2- continued**

```
float ADC2Reading;
float ADCReading;
float FullScale = 5.000;
float Resolution = 2048.000;
/* Channel 1 ADC Reading */
outp(0x078, 0x11);
OSTimeDlyHMSM(0, 0, 0, 1);
msb = (inpw(0x79));
    lsb = (inpw(0x78));
ADCReading = (((msb | lsb) *FullScale)/Resolution);
sprintf(s, "CHA %4.2f",ADCReading );
OSSemPend(LCDSem, 0, &err);
DispStr(1, 0, s);
OSSemPost(LCDSem);
/* Tx CH1 reading to Host Computer */
ps = s;
while (*ps != NULL) {
CommPutChar(COMM1,*ps, OS_TICKS_PER_SEC);
OSTimeDly(5);
ps++;
}
/* Channel 2 ADC Reading */
outp(0x078, 0x12);
OSTimeDlyHMSM(0, 0, 0, 1);
msb = (inpw(0x79));
    lsb = (inpw(0x78));
ADC2Reading = (((msb | lsb)*FullScale)/Resolution);
sprintf(s, "CHB %4.2f",ADC2Reading );
OSSemPend(LCDSem, 0, &err);
DispStr(1, 31, s);
OSSemPost(LCDSem);
ps = s;
/* Tx CH2 reading to Host Computer */
while (*ps != NULL) {
CommPutChar(COMM1,*ps, OS_TICKS_PER_SEC);
OSTimeDly(5);
ps++;
}
/* Read Digital Inputs 1-4 */
DigitalInput1 = (inp(0x77));
sprintf(s,"INP %d\n", DigitalInput1);
OSSemPend(LCDSem, 0, &err);
    DispStr(1, 15, s );
    OSSemPost(LCDSem);
ps = s;
/* Tx DI 1-4 to Host Computer */
while (*ps != NULL) {
CommPutChar(COMM1,*ps, OS_TICKS_PER_SEC);
OSTimeDly(5);
ps++;
}
}
```

**Listing 3**

```
void Task3 (void *data)
{
    UBYTE err;
    DispInit(2,40);
    DispClrScr();
}
```

Continue

Circuit Cellar, the Magazine for Computer Applications. Reprinted by permission. For subscription information, call (860) 875-2199, [subscribe@circuitscellar.com](mailto:subscribe@circuitscellar.com) or [www.circuitcellar.com/subscribe.htm](http://www.circuitcellar.com/subscribe.htm).

**Listing 3—Continued**

```
data = data;

for (;;) {
    OSSemPend(LCDSem, 0, &err);
    /* Get Semaphore for LCD */
    DispStr(0, 13, "MicroC/OS-II");
    /* Display Msg on LCD */
    DispStr(0, 38, "|");
    /* Display spinning wheel on LCD */
    OSTimeDly(10);
    DispStr(0, 38, "/");
    OSTimeDly(10);
    DispStr(0, 38, "-");
    OSTimeDly(10);
    DispStr(0, 38, "|");
    OSTimeDly(10);
    /* Release Semaphore */
    OSSemPost(LCDSem);
}
}
```