Micriμm, Inc.

μC/OS-II

and

The JK microsystems μFlashTCP

Application Note

AN-1010

Jean J. Labrosse
Jean.Labrosse@Micrium.com
www.Micrium.com

Summary

This application note describes how to use µC/OS-II with the JK microsystems µFlashTCP board.

µFlashTCP

The µFlashTCP development kit is an ideal platform for evaluating µC/OS-II and is available from JK microsystems for only $199.95. The µFlashTCP controller (see Figure 1) is a DOS based single board computer powered by an Intel 80386EX microprocessor running at 25 MHz. The µFlashTCP is equipped with a 10Base-T Ethernet port, two serial ports (1-RS-232C and 1-RS232C/RS485), 10 digital I/O lines, 512K SRAM and 512K Flash memory, and a socket for expanding non-volatile memory using M-Systems DiskOnChip products – all in a package slightly larger than a credit card.

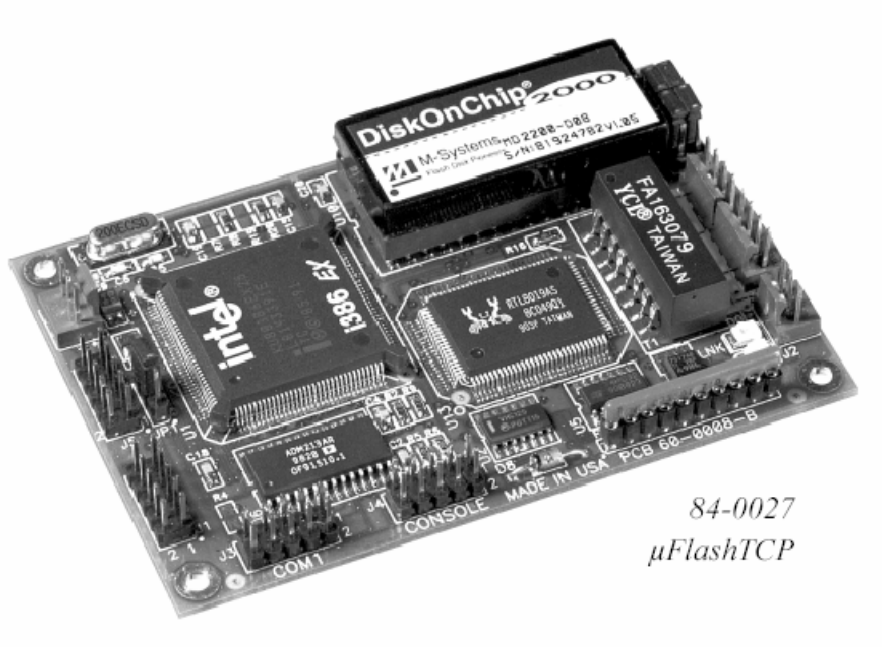

Figure 1, JK microsystems µFlashTCP.

Software development for the µFlashTCP is remarkably simple and quick. Programs are written on a PC compatible computer using C. After your application has been compiled or assembled and linked into a .EXE file (DOS executable file). The .EXE file is then uploaded via one of the two serial ports on the µFlashTCP to the µFlashTCP's flash disk using your favorite telecommunication program (Hyperterminal on a Windows PC) using the X-Modem protocol. The application is then executed on the µFlashTCP and can use the serial port as an output device.

The µFlashTCP is configured with two 'disk drives', A: and B:. Drive A: contains DOS (Disk Operating System), the BIOS (Basic I/O System), and utility programs essential to the operation of the µFlashTCP. Drive A: is read-only. Drive B: is read/write and contains optional utility programs and is available for user files and applications.

The serial port commonly known as COM2 on the PC is the 'console' port on the µFlashTCP. The console is configured at 9600 baud, 8 bits, 1 stop bit and no parity. The baud rate of the console can easily be changed to support baud rates as high as 56K baud.

It is assumed that you will follow the directions provided by JK microsystems to install the µFlashTCP.

Borland C/C++ V4.52

The JK microsystems µFlashTCP comes with a CD that includes the Borland C/C++ compiler V4.52 as well as the TASM assembler. It is assumed that you already installed the Borland C/C++ compiler that comes with the µFlashTCP on your computer which, for sake of testing, I assumed is installed in the `C:\BC45` directory. However, you should be able to modify the build files in case you decided to install the compiler in a different directory.

Windows PC Development Platform

The Borland C/C++ compiler, assembler and linker work just fine on just about any Windows-based PC (Windows 95, 98, NT, 2000 or XP). The compiler assumes that you will open a 'DOS-box' and invoke these tools from within this DOS-box.

µC/OS-II

To run the example, you will need to have µC/OS-II V2.61 (or higher) on the disk drive of your choice. µC/OS-II V2.61 is available as an upgrade from www.Micrium.com (see Products – Purchase and go to the online store).

uCOS-II-uFlashTCP.ZIP

The example code accompanying this application note is provided in the file `uCOS-II-uFlash-TCP.ZIP`. When unzipped, you should have the following files/directories:

`\SOFTWARE\uCOS-II\JKMicro-uFlashTCP\Ex1L\SOURCE`

`INCLUDES.H`	This is the 'master' include file which is expected by µC/OS-II. It assumes that we will be using the 80x86 port files for the 'large memory model'.
`OS_CFG.H`	This is the µC/OS-II configuration file which establishes which options of the OS are being used. You should note that a lot of the µC/OS-II options have been 'compiled-out' in order to reduce the size of the executable. You can simply re-enable the options you need by setting the appropriate `#define` value back to `1`.
`TEST.C`	This file contains the sample code which is similar to Example #1 in the book except that it's been adapted to the JK microsystems µFlashTCP.
`TEST.LNK`	This is the Borland TLINK linker command file which specifies which files are to be linked together in the final load image, `TEST.EXE`.

`\SOFTWARE\uCOS-II\JKMicro-uFlashTCP\Ex1L\TEST`

`MAKETEST.BAT`	This is the batch file that invokes the 'make' utility to build the example code. You simply need to type 'MAKETEST' at the DOS prompt.
`TEST.EXE`	This file is the pre-compiled and linked example. You can simply download this file to the µFlashTCP and run it. In fact, you can even run this code directly in the DOS-box from this directory.
`TEST.MAK`	This is a 'makefile' which contains compiler, assembler and linker commands to build the example executable.

Running the Code

To run (i.e. execute) the example code, simply download TEST.EXE from the 'TEST' directory into the µFlashTCP's B: drive and type 'TEST' followed by the 'Enter' key at the B:\> prompt as shown in Figure 2.

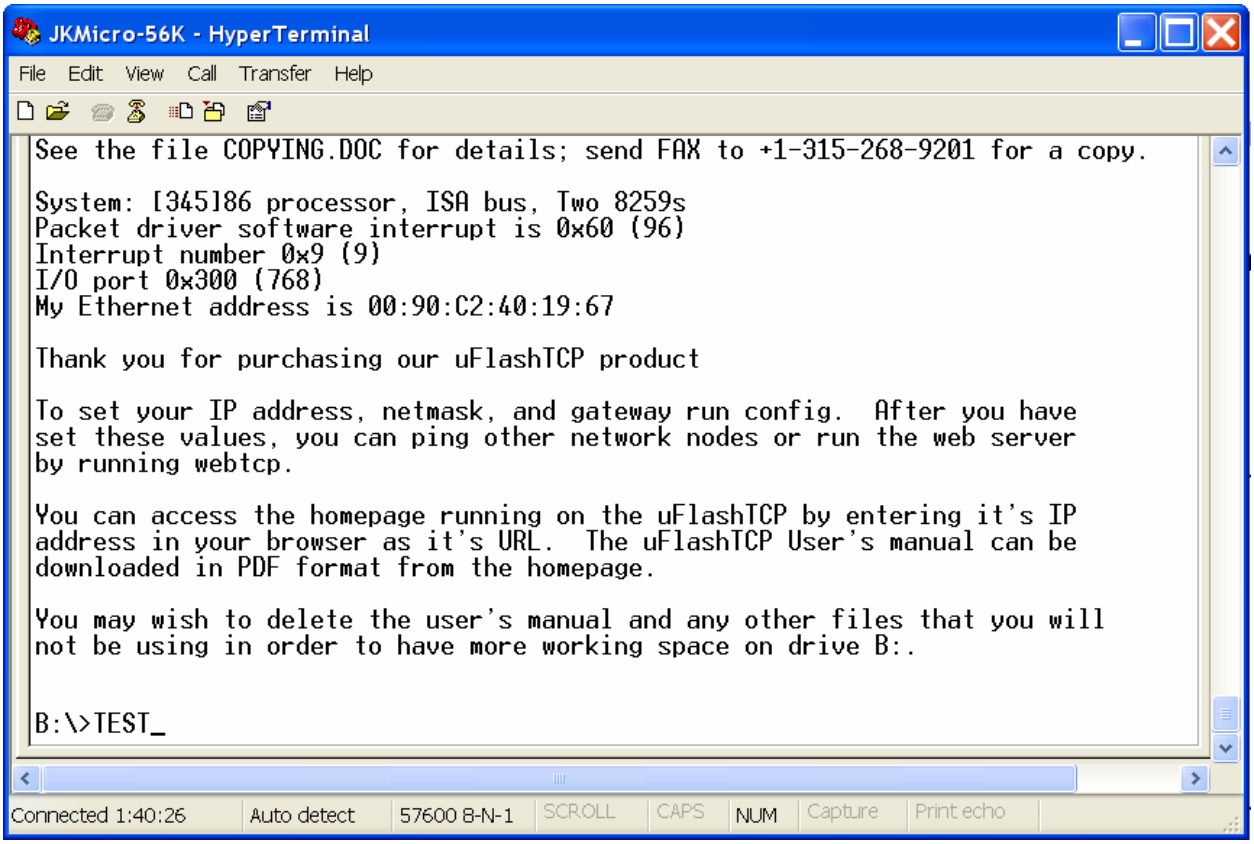

Figure 2, Starting the example code on the µFlashTCP.

Figure 3 shows you the first few seconds of execution.

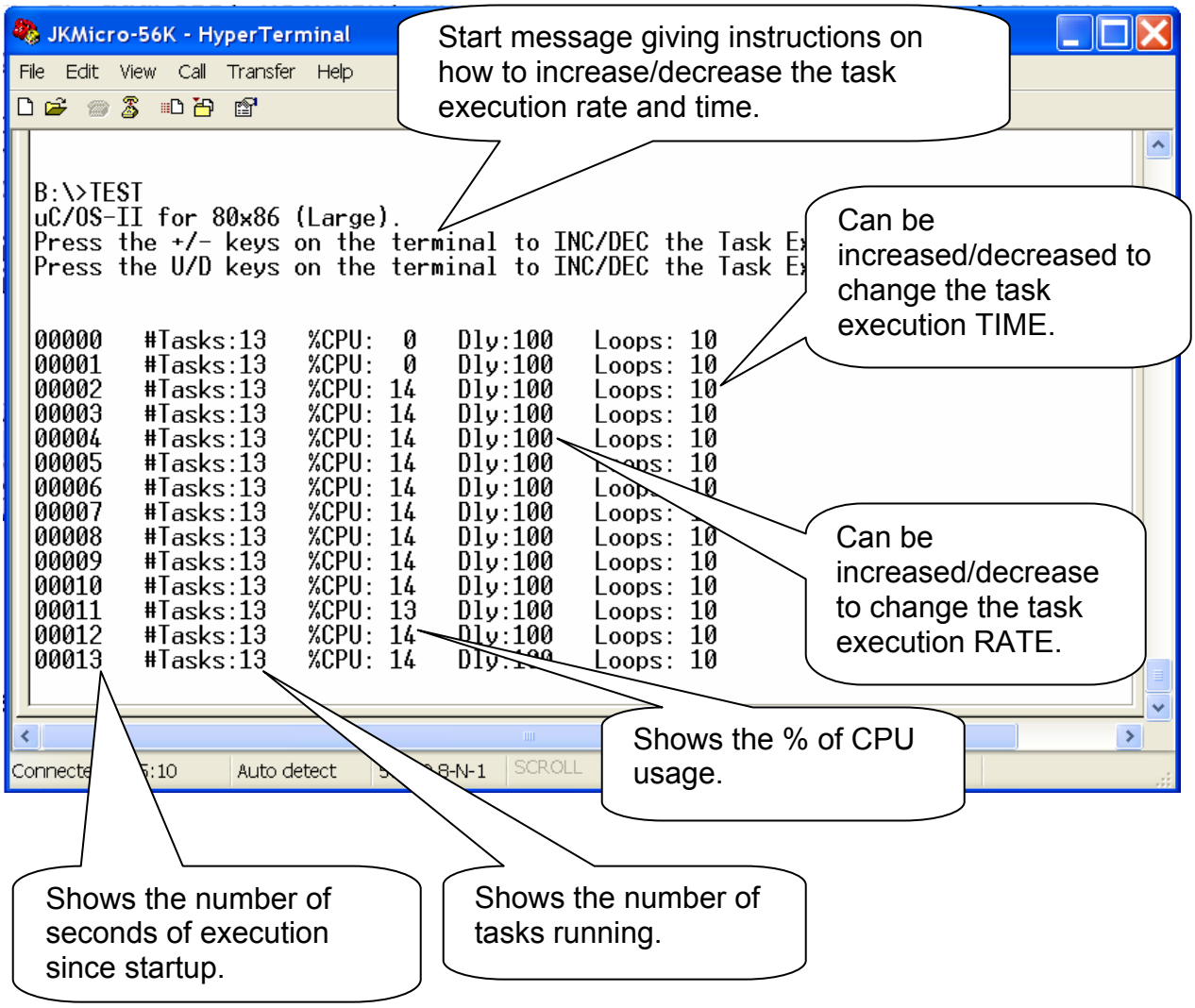

Figure 3, Running TEST.EXE on the µFlashTCP.

Figure 4 shows how the CPU usage is affected by increasing the task execution rate or the task execution time.

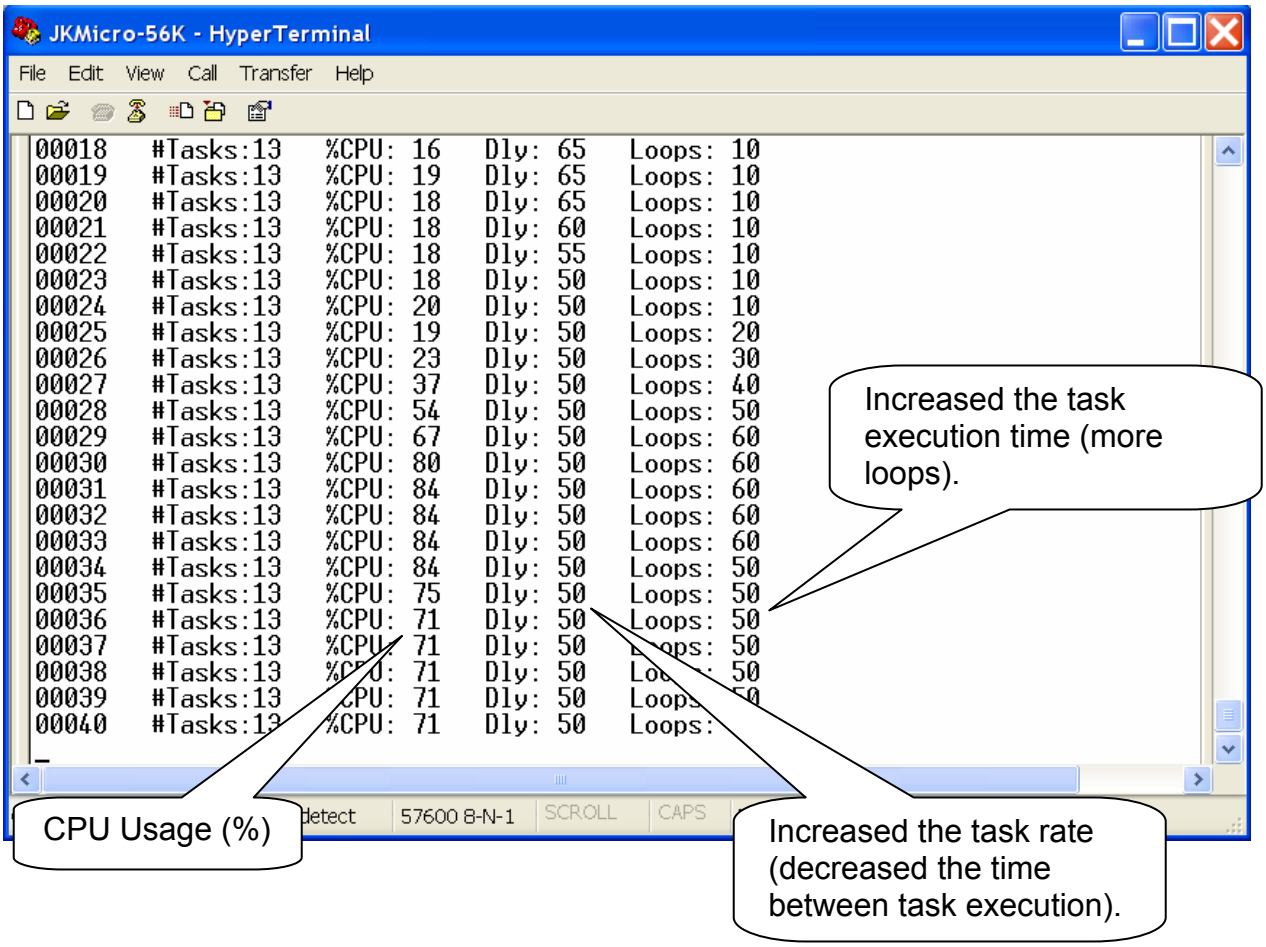

Figure 4, Increasing the task rate and execution time.

Example Code

The example code accompanying this application note is very similar to the code presented in *Example #1* of the µC/OS-II book except that the µFlashTCP doesn't come with a VGA controller and thus, cannot display characters at random locations on the screen. Instead, input and output are done through the µFlashTCP console port (COM2). This is the same port that is used to load the application code in the µFlashTCP.

The test code creates ten tasks that initially execute every half a second. Each task is identical and shown in Listing 1 below.

```
static  void  TestTask (void *pdata)
{
#if OS_CRITICAL_METHOD == 3                    /* Allocate storage for CPU status register           */
    OS_CPU_SR  cpu_sr;
#endif
    INT8U      x;
    INT8U      y;
    INT8U      err;
         INT16U     i;


    pdata = pdata;                             /* Prevent compiler warning                           */
    while (TRUE) {
        OSSemPend(TestRandomSem, 0, &err);     /* Acquire semaphore to perform random numbers        */
        for (i = 0; i < TestLoops * 5; i++) {
            x = random(80);                    /* Waste time by getting a random number              */
            y = random(10);
        }
        x = x;                                 /* Prevent compiler warning                           */
        y = y;
        OSSemPost(TestRandomSem);              /* Release semaphore                                  */
                                               /* Display the task number on the screen              */
        OSTimeDly(TestDly);
    }
}
```

Listing 1, Task code for the 10 identical tasks.

The task doesn't do anything useful except consume CPU time. In fact, you can actually adjust the amount of time consumed by these tasks by pressing either the +/- keys on the console keyboard or the U/D keys.

The '+' key increases the task execution rate by reducing the `TestDly` variable by `5` each time the key is pressed.

The '−' key decreases the task rate by increasing `TestDly` by `5` each time it's pressed.

The '`U`' key increases the `TestLoops` variable and thus consume more time each time the task executes.

The '`D`' key performs the reverse action.

To quit the test code and return to the DOS prompt of the µFlashTCP, simply press the `ESC` (Escape) key on the console terminal.

Conclusion

Providing you a 'shell' allows you to start experimenting with the µFlashTCP. You can add additional tasks and start using the digital I/Os.

JK microsystems also sells a companion board to the µFlashTCP called the µ-IO which is shown in Figure 5. The µ-IO is available for $99.95 to Micriµm customers.

The µ-I/O expansion board gives users the ability to add analog inputs and ouputs, digital inputs and isolated high current outputs to the µFlashTCP. Each µ-IO board adds 4 channels of 12 bit A/D, 2 channels of 12 bit D/A, 4 output drivers and 8 digital inputs. The board installs easily to the back of the µFlashTCP with a short ribbon cable. If more I/O is required, several boards can be added to the system. A library of C functions is supplied to eliminate the hassle of programming devices on a serial bus.

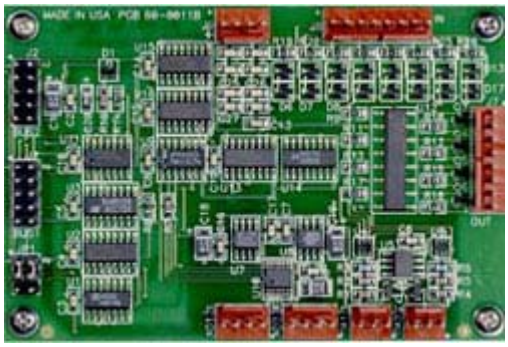

Figure 5, JK microsystems' µ-IO.

References

µC/OS-II, The Real-Time Kernel, 2nd Edition
Jean J. Labrosse
R&D Technical Books, 2002
ISBN 1-57820-103-9

Contacts

Micriμm, Inc.
949 Crestview Circle
Weston, FL 33327
+1 954 217 2036
+1 954 217 2037 (FAX)
e-mail: Jean.Labrosse@Micrium.com
WEB: www.Micrium.com

JK microsystems, Inc.
1403 Fifth Street, Suite D
Davis, CA 95616
+1 530 297 6073
+1 530 297 6074 (FAX)
e-mail: JKMicro@JKMicro.com
WEB: www.JKMicro.com

CMP Books, Inc.
1601 W. 23rd St., Suite 200
Lawrence, KS 66046-9950
USA
+1 785 841 1631
+1 785 841 2624 (FAX)
WEB: http://www.rdbooks.com
e-mail: rdorders@rdbooks.com