μFlashTCP

User’s Manual

Unless otherwise noted, this document and the information herein disclosed are proprietary to JK microsystems, Inc. Any person or entity to whom this document is furnished or having possession thereof, by acceptance, assumes custody thereof and agrees that the document is given in confidence and will not be copied or reproduced in whole or in part to meet the purposes for which it was delivered.

The information in this document is subject to change without notice, and should not be construed as a commitment by JK microsystems, Inc.
JK microsystems, Inc. will make every effort to inform users of substantive errors.

JK microsystems, Inc. disclaims all liability for any loss or damage resulting from the use of this manual or any software described herein, including without limitation contingent, special, or incidental liability.

JK microsystems, Inc. recognizes our customer's need for a consistent product and will make every effort to provide one. In order to provide the best possible product for all of our customers, we reserve the right to make incremental improvements in our product designs.

µFlashTCP is a trademark of JK microsystems, Inc. MS-DOS is a registered trademark of Microsoft Corporation. XDOS is a copyright of HBS Corporation. All other brand and product names are trademarks or registered trademarks of their respective companies.

µFlashTCP User's Manual Version 1.1

Printed in U.S.A
Document Part No. 94-0007
Published June 2002

Limited Warranty

JK microsystems, Inc. warrants each µFlashTCP to be free from defects in material and workmanship for a period of 90 days from the date of purchase. This warranty shall not apply to any unit which has been subject to misuse, neglect, accident, or abnormal conditions of operation.

JK microsystems’ obligation under this warranty is limited to repairing or replacing, at JK microsystems’ option, any unit returned to the factory within 90 days of the date of purchase, provided that JK microsystems determines that the unit is defective and has been used in compliance with the terms of this warranty. If the failure has been caused by misuse, neglect, accident, or abnormal conditions of operation, repairs will be billed at a nominal cost.

The foregoing warranty is exclusive and in lieu of all other warranties, expressed or implied, including, but not limited to, any warranty of merchantability or fitness for any particular purpose. JK microsystems shall not be liable for any special, incidental or consequential damages, whether in contract, tort, or otherwise.

Important Notice

Life Support / Mission Critical Applications

This product is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of our hardware or software could lead directly to death, personal injury, or severe physical or environmental damage.

Table of Contents

Overview

The µFlashTCP single board computer is based on the Intel 386Ex microcomputer. The 386Ex is a high performance, 32-bit, single-chip microcomputer that is software compatible with the Intel 80386 family of microprocessors. Onboard Ethernet provides a direct connection to 10BASE-T networks. DOS compatibility allows development in a familiar environment with a wide range of tools. High endurance flash memory eliminates EPROM programming without worry of damaging the onboard non-volatile memory with repeated program cycles. Applications are uploaded directly into the flash disk. Expansion options provide high capacity flash storage eliminating the size and reliability problems associated with electro-mechanical storage devices.

Software development for the µFlashTCP is remarkably simple and quick. Programs are written on a PC compatible computer in the language of your choice. After your application has been compiled or assembled and linked into .EXE or .COM form, it is uploaded to the µFlashTCP's flash disk with your favorite telecommunications program using the X-Modem protocol. The application can then be tested and debugged through the console serial port. When the application is running to your satisfaction, the startup batch file can be modified so that the application will load and execute upon reset or powerup.

These features yield a quick and cost effective solution for applications such as networking, embedded web and serial protocol conversion.

Features

25MHz Intel 386Ex Processor
10BASE-T Ethernet Controller, NE2000 Compatible
5 Volt DC power
512k Bytes RAM Memory
512k Bytes Flash Memory
High Speed PC Compatible Serial Ports:
    1 Full-Function (8-wire) RS-232 Port
    1 Software Configurable as 3-wire RS-232 or RS-485
Synchronous 4-Wire Serial Port
3 PC Compatible Counter/Timers
Processor Watchdog (Generates board wide, hardware reset)
10 Digital I/O Lines
32Pin Dip Socket to accept 512k x 8 bit SRAM, 512k x 8 bit Flash, or
    M-Systems DiskOnChip 2000.
Support for Multi-I/O Expansion Boards
DOS and BIOS Compatible with the Flashlite386Ex and FlashTCP
Compact Size, 3.75" x 2.50" (95mm x 63.5mm), 1.6oz (45gm)

Operation

The µFlashTCP is configured with two ‘disk drives’ A: and B:. Drive A: contains the operating system, the BIOS, 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.

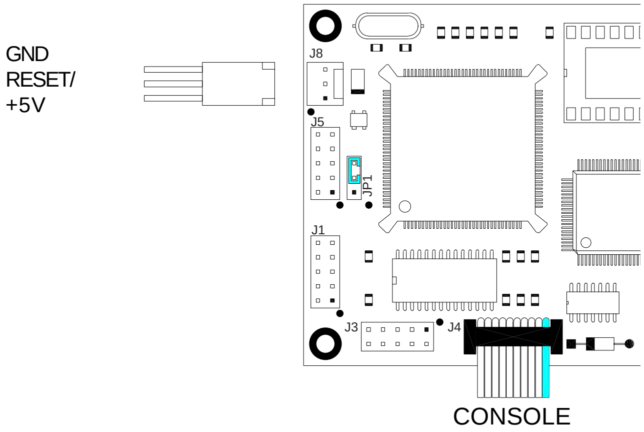
The serial port commonly known as COM2 on the PC is the console for the µFlashTCP. The port is configured for 9600 baud, 8 data bits, 1 stop bit and no parity. This is the primary mode of communicating with the µFlashTCP. DOS and the BIOS treat the console port as the logical devices STDIN and STDOUT. The second port is addressed and assigned interrupt vectors the same as COM1 on a PC.

When power is applied to the µFlashTCP, or when it is reset, the board goes through its initialization procedure and then starts DOS. A simple `AUTOEXEC.BAT` file is executed and then the board is ready to use. The batch file performs several functions before the user is given control. The DOS search path is set, the DOS prompt is set, the CNTL-C flag (discussed later in this manual) is checked and finally, an attempt is made to execute a file named STARTUP on the B: drive. This provides a convenient way for custom applications to execute immediately after initialization of the µFlashTCP. If you wish to have your application start automatically, create a batch file named `STARTUP.BAT` that invokes the program. It is possible, but not recommended, to rename your application `STARTUP.EXE` or `STARTUP.COM`. If this is done and the program locks up, typing CNTL-C at bootup may not break the program and exit to the DOS prompt.

! Although the flash memory devices used have a guaranteed lifetime of over 10,000 write cycles, it is possible for an application to quickly wear them out. The flash memory is intended to store programs and setup data which is normally not changed. Avoid storing data or frequently changed information on the flash disk.

Getting Started

To begin development with the µFlashTCP, you will need a PC compatible computer with a telecommunications program and a free serial port. Connect the µFlashTCP's connector J4 to the PC's serial port with a 9-pin ribbon cable, PN 86-0000. Run the telecommunications program and configure the serial port for 9600 baud, 8 data bits, 1 stop bit and no parity. Apply power to the µFlashTCP, using our A/C adapter PN 88-0004 or a source of regulated 5V DC capable of supplying 400mA. J8 pin 1 (square pad on bottom of board) is positive.



The µFlashTCP should respond with a welcome message and a B: prompt. Enter `DIR` to look at the directory of drive B:. If you do not get a welcome message or echo of the characters that you type, you need to check your serial port setup. To test everything but the µFlashTCP, remove the serial cable from J4 and jumper pins 3 and 5 with a wire or paper clip. If characters typed on the keyboard are not echoed on the screen, the problem is with your setup. You must resolve the problem before you can continue.

If you were able to do a `DIR`, take a few minutes to explore the contents of the µFlashTCP's file system. You will find all of the essential utilities on drive A: and some optional programs on drive B:. Drive A: is write-protected and cannot be altered. Drive B: is read/write and can be changed or reformatted.

After you have looked at the programs on the µFlashTCP, the next step is to try to upload a file. This is the procedure for getting a file from your PC to the µFlashTCP. On the µFlashTCP, type the command UP followed by the name of the file you wish to upload. The µFlashTCP will begin sending characters to your PC polling it for the file.

On your PC, start the transfer, usually by pressing the PgUp key. The telecomm program should respond by requesting the file name and protocol. Enter the file name and select X-Modem for the protocol. The transfer should start and when it is complete, you should get a new B: prompt on the screen. If the transfer does not work, the problem is most likely the Carrier Detect signal (pin 1 on the DB-9 connector) into the PC being sensed as low or false. Make sure that the signal is at least +3 volts into your PC if you are not able to transfer files.

If the transfer terminated without problems, you have a working development environment for the µFlashTCP controller. At this point, you may wish to download the files `EDIT.COM` and `BASIC.COM` from the µFlashTCP to your PC. Start the download on the µFlashTCP by typing DOWN BASIC.COM and pressing Enter. On your PC, begin the transfer, usually by pressing PgDn. After the file is transferred, repeat the process with `EDIT.COM`. These files are also found on the development CD and JK microsystems website.

The µFlashTCP does not have a hardware clock calendar. The time and date will be maintained by DOS until the power is cycled. If the correct time and date are required after a power cycle, they must be set either from the command prompt or via a server on the network. Time and date can be set with the following commands:

`B:\>TIME 13:30:00`	Sets the time to 1:30 pm
`B:\>DATE 4-19-2002`	Sets the date to April 19, 2002

The console output and input on the µFlashTCP can be controlled with the `QUIET` and `NOQUIET` commands. This is useful for applications where both serial ports must communicate with hardware devices that would be disturbed by console messages. Before we look at the `QUIET` and `NOQUIET` commands, an overview of the µFlashTCP boot procedure might be helpful.

When power is applied to the µFlashTCP, one of the first things the BIOS initialization code does is check for a CNTL-C character typed at the console. If this character is received as soon as the board is powered up or reset, a flag is set which overrides the quiet state of the console. When DOS runs its `AUTOEXEC.BAT` file on drive A:, the state of the CNTL-C flag is also checked and any user applications set to run on drive B: are not loaded. This insures that a hung application or quiet console can always be interrupted. Note that the CNTL-C break must be transmitted to COM2 and 9600 baud, regardless of the board configuration. This provides a consistent method of interrupting the boot process.

Running QUIET will turn off both input and output on the console port, allowing applications to use it as COM2. Transmitting a CNTL-C immediately after reset or powerup will restore the console until the next reboot. Running NOQUIET will restore the default setting of an active console. If COM2 is receiving a constant stream of binary data that contains the CNTL-C (ASCII 0x03) character, you may experience trouble booting even in QUIET mode. Please contact JK microsystems for more information regarding this situation and possible solutions.

Some applications require switching console states on the fly. The quiet flag is stored in the flash BIOS and is copied into RAM when the board boots. Once in RAM, the flag is located at 40:008Bh and occupies the entire byte. Writing a 0 to this location will enable the console, a 1 will disable the console. The C code below illustrates this process:

```
char far *c;
c=(char far *)MK_FP(0x40,0x8B);
*c = 0;                         /* turn on console */
*c = 1;                         /* turn off console */
```

If the CNTL-C flag is not set, the AUTOEXEC file will attempt to transfer control to a file named STARTUP on drive B:. DOS also looks for and, if present, loads CONFIG.SYS from drive B:.

A µFlashTCP in quiet mode may appear to be non-functional. When troubleshooting a system, always try pressing CNTL-C while applying power.

Hardware

Memory Configuration

The 386Ex processor is initially configured in real mode with a physical address space of 1 megabyte. The SRAM is located between 00000h and 7FFFFh, the flash is between 80000h and FFFFFh. A 32-pin DIP socket is provided for additional flash, RAM, or EPROM data. This memory can be accessed by reprogramming the chip select unit in the 386Ex or by entering protected mode.

During the boot process the BIOS is copied from flash into the top of RAM. The BIOS executes out of RAM. After the BIOS is copied out of flash, the flash is removed from the memory map with the exception of a small window near the 1 meg boundary. This allows the reset procedures to work properly while maintaining user access to peripherals mapped in the higher portion of memory. When a request for data on drive B: is processed, the flash is mapped in, the drive read, then mapped out again. If present, the DiskOnChip occupies a block of memory starting at segment E000 hex.

I/O Configuration

The 386Ex is configured for enhanced DOS mode. This mode provides access to the PC/AT peripherals such as UARTs, counter/timers, and the interrupt controller at their traditional I/O port addresses. Other 386Ex peripherals are accessible in expanded I/O space.

For addressing and programming the peripherals specific to the 386Ex, please refer to the Intel 386Ex Embedded Microprocessor User’s Manual (Intel document number 272485-002). The manual is available in PDF format from our web site at http://www.jkmicro.com

Clocks

The 386Ex is fed with three clock signals. The processor clock of 50 MHz, a baud rate clock of 1.8432 MHz and the Timer clock at 1.1932 MHz. The 1.1932 MHz timer clock feeds the input of the processor’s three timer channels. Timer 0 is used for the DOS timer tick and is generally not changed. The two remaining timers are free for use with applications. These timers can be clocked either by the external 1.1932 MHz timer clock or by the internal prescaled clock (PSCLK), depending on the application requirements.

Digital I/O Ports

The µFlashTCP has 2 ports controlling a total of 10 bits of I/O.

386Ex Port 1 bits 4,5,6 and 7, I/O Address F860 and F862 hex
These signals are available on J5. The data on Port 1 can be read from I/O address F860 hex. The default configuration is input. Each bit of Port 1 can be individually configured as an input or output. To configure a bit for output, write a zero in that bit position to I/O address F864 hex. To output data on Port 1, write the data to I/O address F862 hex.

P1PIN: F860h, Port Pin Status Register (read only), bits 4-7
P1LTC: F862h, Port Latch Register, bits 4-7
P1DIR: F864h, Port Direction Register, bits 4-7, 0 for output, 1 for input or open drain output.
P1CFG: F820h, Port Configuration Register, bits 4-7 low, route P1.4-P1.7 to chip pins (BIOS Default)

386Ex Port 3 bits 0-5, I/O Address F870 and F872 hex
Port 3 bits 0,1,2 and 5 are available on J1, bits 3 and 4 are available on J5. The data on Port 3 can be read from I/O address F870 hex. The pins default to inputs. Each bit of Port 3 can be individually configured as an input or output. To configure a bit as an output, write a zero to that bit position in I/O address F874 hex. To output data on Port 3, write the data to I/O address F872 hex. When used as inputs, these pins can also be configured to generate processor interrupts.

P3PIN: F870h, Port Pin Status Register (read only), bits 0-5
P3LTC: F872h, Port Latch Register, bits 0-5
P3DIR: F874h, Port Direction Register, bits 0-5, 0 for output, 1 for input or open drain output.
P3CFG: F824h, Port Configuration Register, bits 0-5 low, route P3.0-P3.5 to chip pins (BIOS Default)

Be sure to use a read-modify-write process when changing the I/O ports. Pins on both ports are used to control other on-board functions that can be reprogrammed or disabled through these configuration registers. See the Intel documentation and the board schematic for more information on configuring these ports.

Programming the Ports

The I/O ports on the µFlashTCP are mapped into the 386Ex I/O space. Using the ports requires the use of functions unique to the x86 family of processors. Creating a pointer to the location may seem logical, but that reference would be in memory space, not I/O space. Borland C functions `inport(`*`port`*`)` and `outport(`*`port,value`*`)` are 16 bit (word) instructions, `inportb(`*`port`*`)` and `outportb(`*`port,value`*`)` are 8 bit (byte) instructions. These functions are part of the `dos.h` header file. Similar functions (and header files) are available for other C compilers and languages. The following code illustrates the use of `inportb()` and `outportb()`.

```
unsigned char port;
port = inportb(PORT_DIR);          /* get value of dir. reg */
port |= PORT_DIR_MASK;             /* set dir. bit for input */
outportb(PORT_DIR, port);          /* write value to dir. reg */
printf("PORT: %X\n", (int)inportb(PORT));
                                   /* read & print port value */
```

Asynchronous Serial (COM) Ports

The µFlashTCP has 2 serial ports, COM1 and COM2. Both ports are internal to the 386Ex and are compatible with the 16450 UARTs on a PC. The maximum data rate is 115k Baud.

COM1 is wired as Data Terminal Equipment (DTE) for connection to a peripheral such as a modem. This is a full function RS-232 port implementing all of the handshaking and control lines with the exception of the Ring Indicator input. See Table 2c for the connector wiring. The UART base address is at I/O location 3F8h and can be configured to use IRQ 4. This port can also be configured to use a DMA channel. See the utilities CD or our website for example code.

COM2 is the default console and is wired as Data Communications Equipment (DCE) for direct connection to a computer or terminal. This port is software configurable as a 3 wire RS-232 port implementing RxD and TxD or as a half duplex RS-485 port. See Table 2d for the connector wiring. The UART base address is at I/O location 2F8h and can be configured to use IRQ 3.

Table 1 shows the UART configuration and control registers. Please refer to the Intel 386Ex data sheet for more information on the serial ports and their configuration.

The DATA and IER registers also hold the baud rate divisor. When the high bit of the LCR (DLA) is set, the divisor value can be written to DATA and IER. DATA contains the low byte and IER contains the high byte. To determine the required divisor, divide 115200 by the required baud rate. Program the divisor with the nearest integer value. When access to the divisor value is no longer required, clear the DLA bit.

	7	6	5	4	3	2	1	0
Base	Receive/Transmit Holding Register / Divisor Latch Low (DATA)							
	Data In, Data Out							
Base+1	Interrupt Enable Register (IER)							
	0	0	0	0	Modem Status	Receive Line Status	Transmit Buffer Empty	Receive Buffer Full
Base+2	Interrupt Identification Register / Divisor Latch High (IIR)							
	Reserved	Reserved	Reserved	Reserved	Reserved	Interrupt Source 00=Modem Status 01=Transmit Buffer Empty 10=Receive Buffer Full 11=Receiver Line Status		Interrupt Pending (0=Pending)
Base+3	Line Control Register (LCR)							
	Divisor Latch Access	Send Break	Parity 000=None, 001= Odd, 011=Even, 101=Mark, 111=Space			Stop Bits, 0=1, 1=2	Word Length, 00=5, 01=6, 10=7, 11=8	
Base+4	Modem Control Register (MCR)							
	0	0	0	Loop Back Test	Ext. Int. Enable	Out1	RTS	DTR
Base+5	Line Status Register (LSR)							
	Reserved	Transmit Register Empty	Transmit Buffer Empty	Break Interrupt	Framing Error	Parity Error	Overrun Error	Receive Buffer Full
Base+6	Modem Status Register (MSR)							
	DCD	RI	DSR	CTS	Δ DCD	Δ RI	Δ DSR	Δ CTS

Table1: UART Registers

RS-485 Configuration

The COM2/Console port of the µFlashTCP 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 (J3) 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 */
```

The RTS line on COM2 is used to control the RS-485 transmitter. To transmit RS-485 data, set bit 1 of I/O port 2FC hex (mirrored at F8FC hex). To receive RS-485 data, clear bit 1. Note that the state of the chip pin is the inverse of the bit in the register (register=1, pin=0).

```
#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 */
```

The RS-485 driver is internally looped back. Characters transmitted will appear in the UART receiver. Software will need to be written to handle this condition. Two utility programs are available to aid RS-485 development. `485RX` accepts RS-485 data and displays it on the console. `485TX` accepts console data and sends it out the RS-485 port. The console must be set to COM1 when using these utilities. Both programs are installed on drive A:.

Watchdog Timer

The Watchdog Timer is a feature of the 386Ex processor chip. When in use, the watchdog counter decrements once per processor clock cycle. When the counter reaches zero, the WDTOUT pin is asserted for 8 processor clock cycles. This signal can be internally routed to IR7 or routed externally to generate a hardware reset. Software should periodically reload the counter register indicating that it is behaving properly.

The watchdog system has three modes: General Purpose, Software, and Bus Monitor. In General Purpose mode the counter can only be reloaded after it times out. This makes it difficult to use for system protection. The Software mode allows the countdown register to be reloaded before it times out. This allows protection against locked up software. The Bus Monitor mode is not applicable to the µFlashTCP hardware.

To enable the watchdog in Software mode the following sequence must be followed:

1. Write the upper byte of the reload value to the WDTRLDH register (F4C0h)
2. Write the lower byte of the reload value to the WDTRLDL register (F4C2h)
3. Write two sequential words, F01Eh followed by 0FE1h, to the WDTCLR register (F4C8h)

Application software must periodically repeat step 3 to refresh the counter and prevent it from resetting the board. The current value of the counter can be read from WDTCNTH and WDCNTL at F4C4h and F4C6h. Once enabled in Software mode, the timer can not be disabled.

The following C code illustrates the above steps:

```
#define WDTRLDH 0xF4C0
#define WDTRLDL 0xF4C2
#define WDTCLR          0xF4C8

// setup WD timer
// 1.5 sec * 25MHz ->  37.5E6 cycles -> 0x23C3460
outport(WDTRLDH, 0x23C);          // write hi byte of counter value
outport(WDTRLDL, 0x3460);         // write low byte of counter value

outport(WDTCLR, 0xF01E);          // write enable bytes
outport(WDTCLR, 0x0FE1);
```

The watchdog is active on power up, defaults to general purpose mode and the counter has an initial value of 3FFFFF hex. The counter can be disabled by setting bit zero of the WDTEN register at F4CAh. Please refer to the Intel 386EX Embedded Microprocessor User’s Manual for more information.

Ethernet

The Ethernet port is a 16 bit design that supports direct connection to a 10BASE-T network, jumperless configuration, and NE2000 software compatibility. The controller has a default base address of 300 hex and IRQ9, using full duplex twisted pair wiring supporting link detect.

The Ethernet controller requires a software driver to interface with network software or other programs. The supplied packet driver (`NE2000.COM`) configures the chip interrupt, base address and other necessary parameters.

To install the packet driver, type:

```
        B:\> NE2000 0x60 9 0x300
```

The first parameter (0x60) is the software interrupt that programs will use to communicate with the driver. The number could be different, but 60h is common. The driver will install using IRQ9 and I/O base address (300h) and will locate the MAC (Media Access Control) number stored in the on board configuration EEPROM. The user must specify the IRQ and base I/O address used by the controller. When the driver has loaded, it will indicate the system type, software interrupt, port address, IRQ, and MAC number (Ethernet Address).Note that hex values are preceded by ‘0x’, consistent with C programming language syntax.

After the driver has been installed, network software will be able to communicate with the Ethernet adapter and the network. Users will probably want to modify their `STARTUP.BAT` file to automatically load the packet driver.

Other network drivers supporting NE2000 compatible hardware may be used. Drivers for software requiring NDIS or ODI support are available.

The board has two LEDs that indicated the status of the Ethernet link. The LNK LED indicates the status of the Ethernet. When illuminated, the µFlashTCP is receiving the Ethernet ‘heartbeat’ and is connected to a live network. If this LED is not illuminated, there is a problem with the Ethernet wiring or the network. The ACT LED indicates activity on the network. The LED will flash when a data packet is received or transmitted.

DiskOnChip

M-Systems’ DiskOnChip 2000 is a new generation of high performance single-chip Flash Disk. The DiskOnChip has become the standard Flash Disk module for Embedded Single Board Computers. The DiskOnChip MD2000 and MD2800 are Flash Disks in a standard 32-pin DIP package that has built-in TrueFFS (True Flash File System) technology, allowing full read/write disk emulation. TrueFFS provides hard disk compatibility at both the sector and file level. Drive with capacities from 2 to 144 Mbyte are available. Drives larger than 32Mbyte require partitioning to allow XDOS to access the entire drive.

Install the DiskOnChip module in the memory expansion socket U11. Note the location of pin 1. Set the Memory Type jumpers (JP2) for Flash memory. If the DiskOnChip is installed and functioning, there will be an installation message that is displayed during the boot process and a C: drive will be available to DOS.

```
Bios Version 3.3c for uFlashTCP with NE2000 Etherne
DOC Socket Services - Version 0.2
(C) Copyright 1992-1996, M-Systems Ltd.

TrueFFS-BIOS -- Version 3.3.7 for DiskOnChip 2000 (V1.10)
Copyright (C) M-Systems, 1992-1998

DOS Version 3.3c for JK microsystems Flashlite
(C) HBS Corp and JK microsystems 1991-1999

B:\>
```

If your application requires the µFlashTCP to boot from a DiskOnChip, please contact JK microsystems for more information and the configuration procedure.

Jumpers

JP2 - Socket Memory Type

This jumper selects the type of memory in the expansion socket. Available choices are SRAM or Flash. Other memory types may be supported if their pinout is compatible with standard SRAM or Flash chips.
Jumper pins 1-2 and 3-4 for SRAM or pins 1-3 and 2-4 for Flash.
Default position: 1-3 and 2-4, Flash memory expansion.

JP3 - Boot Memory Location

This jumper allows the board to boot from the expansion socket. This is useful when performing field updates of the on-board Flash memory or when using an operating system other than DOS. Jumper pins 1-3 and 2-4 to boot from the on-board memory or jumper pins 3-5 and 4-6 to boot from the expansion socket.
Default position: 1-3 and 2-4, Boot from on-board flash.

Cables and Connectors

The following tables show the pin and signal names (direction) for each connector.

NOTE: N/C indicates no connection and PULLUP indicates a 1k ohm pullup resistor to Vcc. Outputs refer to signals driven by the board and received by a peripheral. Inputs are driven by a peripheral and received by the board.

J1			General I/O
Vcc	1	2	P3.0
P3.1	3	4	P3.2
P3.5	5	6	SSRXCLK
SSTXCLK	7	8	SSRX
SSTX	9	10	GND

Table 2a: General I/O Pinout

J5			Multi-I/O Bus
Data (P1.4)	1	2	CLK (P1.5)
Reset (P1.6)	3	4	CS (P1.7)
Vcc	5	6	GND
IRQ5 (P3.3)	7	8	IRQ6 (P3.4)
Vcc	9	10	GND

Table 2e: Multi-I/O Pinout

J2	RS-485
DATA +	1
GND	2
DATA -	3

Table 2b: RS-485 Pinout

J8	Power
+5V	1
RESET/	2
GND	3

Table 2f: Power Pinout

COM1 is configured as a DTE port, and is generally used to communicate with a peripheral device. COM2 is configured as a DCE port, generally being used to connect the µFlashTCP to another computer. A 10 pin dual row header to 9 pin D-type connector may be required to connect the expansion card to a peripheral or computer. See the tables below for connector pinouts.

J3			COM1:
DCD (in)	1	2	DSR (in)
RxD (in)	3	4	RTS (out)
TxD (out)	5	6	CTS (in)
DTR (out)	7	8	N/C
GND	9	10	N/C

Table 2c: COM1 Pinout

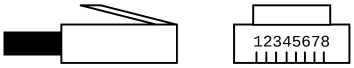
J4			COM2:
PULLUP	1	2	N/C
TxD (out)	3	4	PULLUP
RxD (in)	5	6	PULLUP
N/C	7	8	PULLUP
GND	9	10	N/C

Table 2d: COM2 Pinout

The Ethernet port requires a short adapter (PN 86-0018) to connect between the 8 pin header (J10) and the RJ-45 style connector common to 10BASE-T networks. The default wiring for this adapter allows direct connection to a network hub with a straight through patch cord. If connection to a PC (without a hub) is required, the Tx and Rx pairs can be swapped. This can be accomplished by changing the adapter wiring as outlined below. If external LEDs for link and activity are required, they can be wired to pins 5 through 8 of J10. 680 ohm series resistors are provided on the PCB. Table 2g shows cable wiring and pin numbering information.

J10	Straight Through (to Hub)		µFlashTCP Signal Name	Cross-Over Through (to PC)	
	RJ-45 Pin #	Color (86-0018)		RJ-45 Pin #	Color (86-0018)
1	1	BLU	TxD +	3	BLK
2	2	ORG	TxD -	6	YEL
3	3	BLK	RxD +	1	BLU
4	6	YEL	RxD -	2	ORG
5			LNK LED +		
6			LNK LED -		
7			ACT LED +		
8			ACT LED -		

Table 2g: Ethernet Pinout



Pin numbering on RJ45 cable plug

Pin 1 of a connector can be identified in several ways. Pin 1 has a square PCB pad and the others are round. This should be visible on the bottom of the PCB. Pin 1 will also be identified on the board silkscreen with a ‘1’ and/or a dot. Dual row headers have ODD numbered pins on one side and EVEN numbered pins on the other. The dual row header numbering scheme follows the numbering for an IDC style ribbon cable. This numbering may not be identical to connectors with discrete wires. Use caution when connecting cables to the µFlashTCP.

Applications

The µFlashTCP is compatible with a variety of interface drivers, TCP/IP stacks and network software. Discussion of the installation and operation of these packages is beyond the scope of this manual. Please refer to the software documentation or contact JK microsystems.

Some care is required when setting up the µFlashTCP on a network. Contact the Network Administrator if there are any questions about the required information. When working with a TCP/IP network, obtain the following information before starting your configuration: IP Address, Subnet mask, Name Server Address (DNS), and Gateway address. This information will be required during the configuration process. Other types of LANs require node names, workgroup names, etc. Proceed cautiously, networks can be easily disrupted when nodes are added without careful configuration.

Supported PC BIOS Functions

The Flashlite BIOS supports the following functions (software interrupts) common to PC compatible computers. Please refer to a DOS/PC reference for more information on DOS and BIOS software interrupts.

Int 10h, Video Driver, functions 9 and Eh
Int 11h, Get Equipment Configuration
Int 12h, Get Memory Size
Int 13h, Disk Driver, Functions 0-4
Int 14h, Serial Port Driver, Functions 0-3
Int 16h, Keyboard Driver, Functions 0 and 1
Int 19h, Boot System
Int 1Ah, Real Time Clock Driver, Functions 0-5
Int 1Ch, Hook Timer Tick Interrupt
IRQ0, Timer Tick Interrupt

Interrupts

Working with interrupts within the PC architecture can be a bit confusing. The following information should clarify some of the issues, but it is not a substitute for a detailed DOS/PC programming reference.

The PC architecture has a 1Kbyte Interrupt Vector Table located at 0000:0000. This table holds the 4byte address of each Interrupt Service Routine (ISR) providing a total of 256 interrupts. Most of the interrupts are software interrupts. Software interrupts provide essential services in a ‘known’ location. Hardware interrupt requests have an associated interrupt number and ISR in the vector table. The numbering of hardware interrupts is the largest point of confusion when working with interrupts on the µFlashTCP. The 386Ex processor interrupt pins are referred to as INTx in the Intel documentation. PC programmers are familiar with IRQ (Interrupt Request) numbers and the interrupt numbers also referred to as INT numbers. The correlation of all these numbers is critical when using interrupts. Table 3 identifies the IRQ, 386Ex INT number, function on a standard PC, function on the 386Ex and the interrupt number. You will notice a large gap in interrupt numbers between IRQ7 and IRQ8. This is due to the implementation of two cascaded interrupt controllers and the PC legacy. IRQ0-7 are on the first controller, found in the original PC/XT. IRQ8-15 are on the second controller implemented in the PC/AT. IRQ2 notifies the system when an IRQ is triggered on the secondary controller. Table entries without a 386Ex INT listed are not available on the package pins of the 386Ex.

IRQ	386Ex INT	PC Function	386Ex Function	Int. No. (hex)
IRQ0		System Timer	TCU OUT0	08
IRQ1	INT0	Keyboard	P3.2	09
IRQ2		Cascade		0A
IRQ3	INT8	COM2	SIO1 INT	0B
IRQ4	INT9	COM1	SIO0 INT	0C
IRQ5	INT1		P3.3	0D
IRQ6	INT2	Floppy Disk	P3.4	0E
IRQ7	INT3	Parallel Port	P3.5	0F
IRQ8	INT4	Real Time Clock		70
IRQ9	INT5	IRQ2 (redirected)		71
IRQ10			TCU OUT1	72
IRQ11			TCU OUT2	73
IRQ12	INT6	Mouse		74
IRQ13		Math Coprocessor		75
IRQ14	INT7	Hard Disk		76
IRQ15			WDTOUT	77

Table 3: Hardware Interrupts

Each Programmable Interrupt Controller (PIC) has several control registers and commands, most of which are configured by the BIOS and should not require modification in most applications. The primary PIC is located at I/O address 20 hex and the secondary controller is at I/O address A0 hex.

The first command of interest is the Operation Control Word (OCW). Located at BASE+1, it is the mask byte for the interrupts in that controller. Masked IRQ's have a one in the corresponding bit location, enabled interrupts have a zero. Be sure to only change the mask bits for IRQ's handled by your program. For example, to enable IRQ 4:

```
   outportb(0x21,(inportb(0x21) & 0xEF);
```

The other instruction used is the End Of Interrupt (EOI). This instruction is sent to the appropriate PIC after an interrupt has been processed. The EOI is instruction 20 hex and is sent to the BASE address of the controller, for example:

```
   outportb(0x20,0x20);
```

When servicing an interrupt on the secondary controller (IRQ8-15), it is necessary to send an EOI to both controllers.

Utilities

The µFlashTCP comes preloaded with several utilities to aid system development. These utilities are located on drive A: of the µFlashTCP or the Utilities disk.

UP.COM

This utility facilitates uploading files to the µFlashTCP via the console port using the X-MODEM transfer protocol. The utility requires the user to supply the name of the incoming file. Unless otherwise specified, the file is placed in the active directory of the current drive. Be sure that B: is the current drive or a write-protect error will occur when UP tries to write to the read-only A: drive.

```
B:\>up

Upload file with X-MODEM Protocol
Usage:  up file...
Version 2.0 for JK microsystems Flashlite V25 and 386Ex

B:\>up test.exe

Ready, start X-modem upload now,
 Press CNTL-C to abort...
CCCC
B:\>
```

DOWN.COM

This utility facilitates downloading files from the µFlashTCP via the console port using the X-MODEM transfer protocol. The utility requires the user to supply the name of the file to transmit.

```
B:\>down

Download file with X-MODEM Protocol
Usage:  down file...
Version 1.0 for JK microsystems SBC products

B:\>down test.exe

Ready, start X-modem download now,
B:\>
```

FORMAT.COM

If it becomes necessary to reformat the B: drive, FORMAT provides this function. CAUTION, all information on the drive will be lost during the formatting process.

```
B:\>format
Flashlite FLASH Drive Format Program -Version 3.0
System will reboot after successful format...

 Press 1 to initialize Drive B as 418 KB disk
 Press ESC to exit with no changes

>1
Flash Drive is now formatted
   Rebooting system...
```

EDIT.COM

A simple line editor is included to allow quick creation and modification of batch files or other text files. EDIT is similar to Microsoft’s EDLIN provided in earlier versions of MS-DOS. It allows list, insert, delete, and modify. Upon exit, a backup of the original file is created (filename.BAK) and the edits are saved. If a backup file with the same name already exists, it is overwritten. A list of commands and their usage is available by entering ‘h’ at the edit prompt (>>). The name of the file to edit must be supplied following the command EDIT on the command line.

```
B:\>edit test.bat
FlashLite Line Editor v1.0
Enter h for help

New File: test.bat
>> i
0: @echo Batch file being processed...
       1: mytsr
       2: myapp
       3: ^Z
>> l
0: @echo Batch file being processed...
       1: mytsr
->     2: myapp

>> q
Save before exit (Y,n): y
File Saved
B:\>
```

SETBAUD.COM

The default console baud rate can be changed using the `SETBAUD.COM` utility.

```
B:\>setbaud
Set console baud rate
Usage: setbaud 9600
Possible baud rates: 1200, 2400, 4800, 9600, 19k, 38k, 56k
Version 1.0 for 386Ex based products

B:\>setbaud 9600
Baud rate change will take effect after next reboot

B:\>
```

Note that the CNTL-C break must be transmitted to COM2 and 9600 baud, regardless of the board configuration. This provides a consistent method of interrupting the boot process.

QUIET.COM / NOQUIET.COM

This pair of utilities control the functionality of the console (COM2) port. QUIET will disconnect the DOS STDIN and STDOUT functions from the hardware allowing a user program direct access to COM2. Running this program appears to do nothing. It has no output and just returns to the DOS prompt; no changes are apparent. The next time the board is booted, no output will be present on the console. To restore the console to its active state, boot the board using the CNTL-C process described previously, and then run NOQUIET.

DOS

JK microsystems' controllers use XDOS, a compact operating system for embedded applications. The XDOS command structure is nearly identical to MS/PC DOS version 3.3. The switches for the DIR command have been changed and expanded. XDOS does not support redirected input or output with the use of < and >, but does support pipes (|). None of the external DOS commands are available due to size constraints. XDOS does not support installable file system functions.

XDOS Command Reference

In the list below, XDOS commands are followed by a function description and their format including available parameters and switches. Items in boldface type must be entered. Capitals or lowercase letters may be used. Items in italics are parameters. Those in boldface italics must be entered, those in [ ] are optional. All switches are optional. They are shown as [/X]. Spaces and punctuation are to be included. An ellipsis ... following items means that you may repeat the items as often as needed. Do not enter the ellipsis or the square brackets. Most XDOS commands allow the use of wildcards in filenames and extensions. When wildcards (?=one character, *=any character or characters) are used, the command is executed once for each matching file.

Common parameters are:

[*d:*] drive specification - a letter followed by a colon (:), e.g. A:, if no drive is specified, the default drive is used.

[*path*] the path DOS must take in traveling from one directory to another; directory names are separated by a backslash (\).

[*filename*] up to 8 characters used to name a file.

[.*ext*] a three character extension may be added to a filename; an extension is separated from a filename by a period.

CD / CHDIR

Function: Changes the current directory

Format: **CD** or **CHDIR** [[*d:*]*path*]

COPY

Function: Copies a file, combines two or more files into one file, or transfers data between files and DOS devices

Format: **COPY** [*d:*][*path*]*filename*[.*ext*][*switches*]
[+[*d:*][*path*]*filename*[.*ext*][*switches*]
[*d:*][*path*][*filename*[.*ext*]][*switches*]

Switches: /V - verify the contents of new file
/A - copy file in ASCII format
/B - copy file in binary format

DATE

Function: Displays or changes the current DOS date.
Format: **DATE** [*mm-dd-yyyy*]

DEL / ERASE

Function: Deletes (erases) one or more files from a disk
Format: **DEL** or **ERASE** [*d:*][*path*][*filename*[.*ext*]]

DIR

Function: Lists directory entries
Format: **DIR** [*d:*][*path*][*filename*[.*ext*]][*switches*]
Switches: /a - display file attributes
/b - sort by file size (in bytes)
/d - sort entries by date and time
/f - display entries by alphabetic file name order
/n - display entries in directory order (do not sort)
/s - include system and hidden files in output
/h - display this Help screen (any invalid key)

MD / MKDIR

Function: Creates a subdirectory
Format: **MD** or **MKDIR** [*d:*]*path*

PATH

Function: Specifies directories that DOS is to search when trying to locate executable files
Format: **PATH** [[*d:*]*path*[;[*d:*]*path* ...]]

PROMPT

Function: Sets the DOS system prompt
Format: **PROMPT** [*text*]
Text: Resulting Character(s):

$t	The current time stored by DOS
$d	The current date stored by DOS
$p	The current directory
$v	The version of DOS being used
$n	The default drive
$g	The character >
$l	The character <
$b	The character |
$q	The character =
$$	The character $
$_	Carriage return plus line feed

REN

Function: Renames a file
Format: **REN** [*d:*][*path*]*filename*[.*ext*] *filename*[.*ext*]

RD / RMDIR

Function: Deletes a subdirectory
Format: **RD** or **RMDIR** [*d:*]*path*

TIME

Function: Displays or changes the current DOS time
Format: **TIME** [*hh:mm:ss.xx*]

TYPE

Function: Display the contents of a file
Format: **TYPE** [*d:*][*path*]*filename*[.*ext*]

VER

Function: Displays the DOS version number
Format: **VER**

VOL

Function: Displays the volume label of specified drive
Format: **VOL** [*d:*]

QuickBASIC Console I/O

Some of the code produced by Microsoft QuickBASIC and QuickBASIC Professional compilers does not execute properly on the µFlashTCP. In the case of console I/O, we believe that QuickBASIC is generating code for specific hardware and software not present on the µFlashTCP 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. The second problem is that an `INPUT` statement will not echo the data entered by the user. To work around this problem, we have added a feature which allows the application to enable a console echo function in the BIOS. This feature is enabled by setting the byte at 40:8Ah to a one. Likewise, the local echo is disabled by setting 40:8Ah to a zero.

The following BASIC code demonstrates both of these workarounds:

```
start:
       OPEN “o”, 1, “cons:”        ‘ console output
       PRINT #1, “What’s your name? “‘ get string
       GOSUB echoOn                ‘ turn on local echo
       INPUT name$                 ‘ get the name
       GOSUB echoOff               ‘ turn off local echo
       PRINT #1, “”                ‘ go down a line
       PRINT #1, “Hi , “; name$    ‘ print line and name
       END

echoOn:
       DEF SEG = &H40                    ‘ BIOS data seg
       POKE &H8A, 1                ‘ set local echo flag
       RETURN

echoOff:
       DEF SEG = &H40                    ‘ BIOS data seg
       POKE &H8A, 0                ‘ clear echo flag
       RETURN
```

Specifications

Supply Voltage:	5V +/− 5%
Supply Current:	400mA (nominal)
Operating Temperature:	-20 to +85 °C
Humidity:	5 - 90 % non-condensing

I/O Port Characteristics:

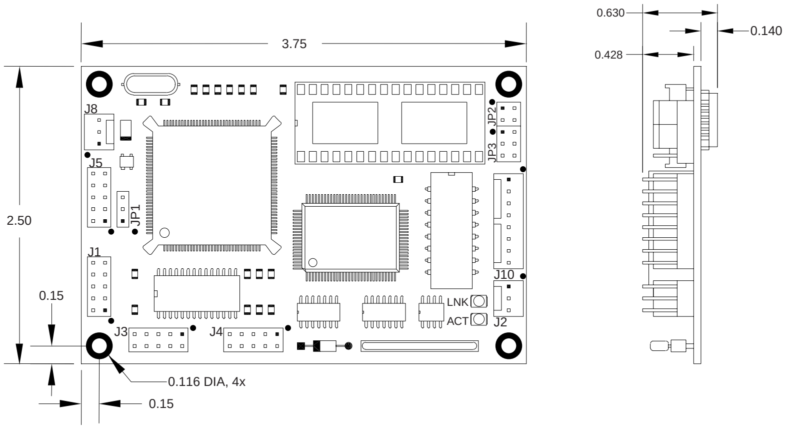
Symbol	Parameter	MIN	MAX	Units	Condition
V_{IL}	Input Low	-0.3	0.8	V	
V_{IH}	Input High	2.0	Vcc+0.3	V	
V_{OL}	Output Low		0.45	V	I_{OL}= 8mA
V_{OH}	Output High	Vcc-0.5		V	I_{OH}= -8mA

Mating Connectors:

Connector	Mfg	MFG P/N	JK micro P/N
J1, J3, J4, J5	Molex	22-55-2101	
	Oupiin	4072-2X05H	28-0030
Pins	Molex	16-02-0096	
	Oupiin	404-PIN-10K	28-0033
J2, J8	Molex	22-01-2031	28-0012
J10	Molex	22-01-2081	28-0037
Pins	Molex	08-50-0114	28-0013

Mechanical:

Dimensions	3.75" x 2.50" (95mm x 63.5mm)
Weight	1.6oz (45gm)



Contact Information

JK microsystems, Inc.
Davis, CA 95616

Telephone:	(530) 297-6073
Fax:	(530) 297-6074
Email:	jkmicro@jkmicro.com
Web:	http://www.jkmicro.com

Intel Corporation
2200 Mission College Blvd.
P. O. Box 58119
Santa Clara, CA 95052

Telephone:	(800) 548-4725, (708) 296-9333
Fax:	(800) 525-3019
BBS:	(503) 264-7999
Web:	http://www.intel.com

M-Systems
DiskOnChip 2000
39899 Balentine Drive, Suite 335
Newark, CA 94560

Telephone:	(510) 413-5950
Fax:	(510) 413-5980
Email:	info@m-sys.com
Web:	http://www.m-sys.com

Dunfield Development Systems
Micro-Basic, Development Tools
P. O. Box 31044
Nepean, Ontario, Canada K2B 8S8

Telephone:	(613) 256-5820
Fax:	(613) 256-5821
BBS:	(613) 256-6289
Email:	info@dunfield.com
Web:	http://www.dunfield.com

```
 Rev      Date      Author           Changes
-----    --------- --------  -------------------------------------------------
 1.1     19JUN02    EW        Add revision log.
                              JP1 not a user jumper, fixed location (2-3).
                              NMI mode no longer supported.
                              Clarify RS485 operation, add examples.
                              Add info on port programming, memory map,
                              interrutps, clocks and supplied utilities.
```