Micro Technology Unlimited
2806 Hillsborough Street
P.O. Box 12106
Raleigh, NC 27605, U.S.A.
(919) 833-1458

MTU-130 COMPUTERS

MTU-130 COMPUTER
OPERATIONAL SPECIFICATIONS
OCTOBER 1, 1981

The MTU goal has been to design the top of the line 6502 GENERAL PURPOSE computer system with strong emphasis on the human interface and system speed. This specification documents the results of achieving this goal. The basic system is titled the MTU-130, which is floppy disk based. MTU recognizes the need for flexibility and offers several configurations which customers can order. The beginning of this document summarizes the major system features and prices, followed by a description of system functions and standard and optional software and hardware. The final section describes system expansion and future modules.

MAJOR FEATURES

* 1MHz 6502 microprocessor with 18 bit addressing for up to 256K clear address space.

* 80K of RAM standard.

* "Keyboard" style package with internal expansion.

* Floppy disk storage options up to 4 Megabytes.

* Video display operating modes:
  * Bit-mapped black and white high resolution graphics in 480 wide by 256 high, allowing 25 lines by 80 characters per line, mixable with graphics
  * Bit-mapped graphics with 4 levels of gray in 240 wide by 256 high

* ALL software is RAM based for upward compatibility

  * CODOS disk operating system with integrated device drivers
  * Software printer drivers for hardcopy of text and graphics or mixed
  * Utility application programs
  * Full screen infinite scrolling MTU-EDITOR

* Standard I/O features:

  * 96 Key keyboard with separate cursor keys and 8 displayed-legend function keys
  * Fiber optic Light pen with high resolution
  * Two 8-bit Parallel ports
  * 1 RS-232 serial port with software baud rate selection
  * 8-bit Digital-to-Analog speech/sound/music port with filter, amp and speaker
  * Cassette tape interface hardware
  * Interface circuitry for a 50K Baud interrupt driven Network option

* 4 ROM or EPROM sockets which can be switched on/off under software control. The system does not use ROM based software, but they are present for specialized applications.

INTRODUCTION

MTU has established a new standard of excellence for the top-of-the-line GENERAL PURPOSE 6502 microcomputer system. The basic design concepts are superior human interface, high speed throughput and user flexibility in system and application software. MTU believes that the user should receive a system powerful enough to perform all necessary functions without having to add memory expansion, graphic expansion, etc. to make the system truly useful. Even with the fast rate of change of technology, MTU expects this system to be built as it is defined here for many years because it answers the true needs of the market, and the level of sophistication built into the software cannot be found on other small systems in the marketplace. Clean system expansion is provided through an 18 bit address bus, internal card file with unused positions and sufficient power supply capacity.

The operating system software uses device independent "channels" and performs many system functions for the user without requiring total user knowledge. For example, the user never needs to worry about whether a disk file is large enough to receive the data or program to be stored to it. The system automatically allocates enough disk space to execute the store.

Additionally, the computer offers a multi-faceted human interface composed of the following components:

1. Audio output of highly intelligible digitized speech, music or sounds.
2. Visual output through a high resolution bit mapped display.
3. Graphic input from the operator through the light pen and cursor keypad.
4. Programmable function keys with legends on the CRT for quick change of presently available user "options" to select from.
5. Software designed for maximum ease of use and growth expansion.

Thus, the system is a powerful computer in its own right, but additionally can readily perform as an intelligent human interface to a larger system or network.

The CRT display technique has been designed to allow maximum flexibility in the design of visual presentations. Using a bit-mapping technique, the display consists of a dot matrix of 480 dot columns by 256 dot rows. This type of format allows any displayed combination of text and high resolution graphics. With the standard operating system driver software, both graphics and text can be combined on the same display. Normally the display has 25 lines by 80 characters with the 25th line (bottom) showing 8 rectangular boxes containing the 8 function key legends. An additional software selectable mode turns the display into a 256 by 240 bit format with four levels of BRIGHTNESS on a per dot basis.

One of the main problems facing the microcomputer user today is the lack of memory address space. This prevents the average user from obtaining additional memory, or performing advanced software functions with the computer. Hal Chamberlin, our V.P. of Research and Development, has solved this problem by designing a new technique for increasing address space with little or no penalty to the programmer or user. Thus, our 18 bit buffered address bus is capable of allowing 256K bytes of memory using a standard 6502 processor. Without available address space, sensible system expansion is nearly impossible.

Another problem the user must face today is that most system software is ROM based. As the market matures, more "standard" features are desired than the manufacturer originally supplied. This creates a problem that the manufacturer either ignores or solves by introducing "new" systems with enhanced ROM software. The latter situation usually means that the new ROM is incompatible with the old, creating early obsolescence of the old systems.

Mainframe systems have long used RAM based software with only a boot ROM for loading the system software from tape or disk during power up. This allows future upgrading of the software or even completely changing it if desired without changing hardware ROMs. Thus maximum flexibility is provided for the user and the market to utilize the full potential of all systems, both new and old.

The approach that MTU has chosen is that software is RAM based. With our high speed disk controller in the computer, program load time is insignificant in most cases. This approach allows a user maximum flexibility, if for example, he wishes only to operate in PASCAL and not in BASIC. Thus, the MTU system is fully controllable under software which can be upgraded instead of becoming obsolete as market needs and desires change.

Given that software is RAM based, a technique is required which would give a software house the ability to protect its products from piracy while allowing the user to make backup copies. MTU has taken a solid approach to solve this problem. Each computer contains a software readable unique encripted serial number. In addition, other numbers can be programmed in by MTU. A unique user number can be assigned and programmed into as many systems as are needed. Thus, a user who wishes to purchase software on a use license basis (e.g., a high school, a university department, a company home office, etc.) may do so. A unique vendor number can also be assigned to a vendor buying from MTU on an OEM basis and selling his customized system to a specific market. We feel that this approach provides the highest benefits to the user in that he has complete flexibility to use and back up the software he has purchased while it allows security for software vendors, thereby stimulating them to write quality software.

To complement the system flexibility, standard I/O hardware/software functions have been chosen carefully to give the basic system the ability to perform real jobs today. Computer users have become more sophisticated over the last several years and require more functions to be performed by their computer system. In this top of the line system, these functions are standard and are accessed through industry standard connectors on the Computer Module. Software authors writing applications for the system can expect all users to have these standard features and use them to the maximum advantage.

The software tools which are provided standard with the system create a productive environment for the user. Both CODOS, and MTU-EDITOR are extremely powerful tools written in machine language for fast operating speeds. Both are usable by computer professionals and neophytes as well. CODOS includes I/O device drivers for text and graphic display, keyboard input and printed output. Additionally it contains 19 utilities for performing frequently needed operations. The structure of CODOS allows the user to easily add new commands, permitting customizing and user interfacing that heretofore has not been expected on personal computers.

At MTU, it was felt that the user needs several levels of interaction with the system hardware and software for interrupting and resetting. Four keys are provided for these functions. Depression of the RESET key performs a WARM RESET. Depression of the MODIFY and the RESET keys together performs a COLD RESET (same as power-on). Depression of the INTERUPT key halts the program which is running and returns control to the system program. Additionally a BREAK key is provided which can be activated by an application program to be ignored or to interrupt when depressed. We feel that all users will come to appreciate these functions.

MTU USER GROUP (MUG)

To promote interchange of software between 130 users, MTU has formed a user group, MUG. Membership for the first year is free with purchase of a 130 or 100 system with prior purchase of a CODOS system. A newsletter will be distributed quarterly to the address on the warranty registration form you return to MTU.

OCTOBER 1, 1981 PRICE LIST (LIST PRICES FIRM THROUGH MARCH 31, 1982)

ORDER NO.	COMPLETE SYSTEMS	LIST *PRICES
MTU-130-1S	MTU-130 WITH 1 SINGLE SIDED DRIVE (.5MBYTES)	3995.00
MTU-130-1D	MTU-130 WITH 1 DOUBLE SIDED DRIVE (1MBYTES)	4195.00
MTU-130-2S	MTU-130 WITH 2 SINGLE SIDED DRIVES (1MBYTES)	4495.00
MTU-130-2D	MTU-130 WITH 2 DOUBLE SIDED DRIVES (2MBYTES)	4995.00

THE ABOVE COMPLETELY OPERATIONAL SYSTEMS CONTAIN:

* STANDARD MTU-130 MODULE
* 80K BYTES OF RAM FOR SYSTEM AND USER SOFTWARE AND DISPLAY
* MTU-BASIC 1.0 WITH GRAPHICS AND DISK LIBRARY EXTENSIONS
* 12" GREEN PHOSPHOR CRT MODULE
* 2 POSITION 8" FLOPPY DISK MODULE WITH 1 OR 2 DRIVES
* ALL CABLES, ETC. -- READY TO PLUG AND RUN

ORDER NO.	INDIVIDUALLY PURCHASABLE ITEMS	LIST
MTU-130	COMPUTER MODULE ONLY, INCLUDING: MONOMEG PROCESSOR BOARD, 80K BYTES OF RAM, FLOPPY DISK CONTROLLER WITH STANDARD SOFTWARE, (SEE THIS BROCHURE), ALL STANDARD I/O CONNECTORS ON BACK PANEL, FIBER OPTIC LIGHT PEN, 96 KEY KEYBOARD, KEYBOARD STYLE CASE, POWER SUPPLY AND FAN, CABLE FOR CRT MONITOR	2640.00

NOTE: THIS UNIT DOES NOT INCLUDE CRT MODULE, FLOPPY DISK MODULE OR POWER TO DRIVE FLOPPY DISK DRIVES.

2200-1M	MTU-130 MANUAL ONLY (credited toward purchase)	50.00
2100-1	MONOMEG PROCESSOR BOARD ASSEMBLY	995.00
2101-1	MTU-BASIC 1.0 WITH 3 STANDARD LIBRARIES	129.00
2101-2	KEYWORD GRAPHIC LIBRARY	49.00
2102-1	MTU INSTRUMENT MUSIC	49.00
2103-1	DISKEX DISK EXCHANGE UTILITY	149.00
2104-1	MTU-ASSEMBLER/DISASSEMBLER	79.00

WARRANTY

All products of MTU are warranted for 6 months (45 days for disk drives) from date of shipment from the factory to be free from defective parts and workmanship. Any other damages including but not limited to: misuse, misconnection to the system, abuse, fire, flood, or other acts of God are not covered by this warranty. Units under warranty requiring repairs are to be returned to the factory postpaid. These will be returned postpaid within 2 weeks from the date of receipt. No statements other than these printed specifications are made or implied. Liability of MTU is limited to repair or replacement of faulty unit(s) and does not extend beyond the purchase price of the unit(s).

MTU reserves the right to change prices, specifications, and availability without prior written notice.

SYSTEM DESCRIPTION

The MTU-130 systems consists of the following hardware subassemblies:

COMPUTER MODULE DUAL FLOPPY DISK MODULE CRT MODULE

MTU sells the systems in various arrangements to meet varying customer needs. These modules are described in full detail below.

MANUALS

System manuals are supplied in a 3 ring binder. Installation, startup and system use procedures, full descriptions and schematics of each hardware board, descriptions and pin connections of each I/O connector, and full user documentation on each standard software program are provided. Additionally, 2 diskettes, filed in the binder, contain the standard software. All optional MTU software manuals will be supplied prepunched to fit into this binder.

COMPUTER MODULE

This Module contains the keyboard, computer electronics and expansion card positions for the system. The Floppy disk signal cable and the CRT Module cable plug into the back of this Module. Switched AC outlets are provided to power the Floppy Disks and CRT when the Computer Module is turned on.

COMPUTER MODULE CASE

The case is designed for attractive appearance while providing maximum protection. The case is 22" wide by 14" deep and 2.5-4" high from front to back. It is constructed of metal for durability and structural integritry. The main surfaces are cocoa brown. The case contains the following subassemblies:

1. Power supply
2. 5 position Expansion Card File
3. System electronic boards - MONOMEG Processor and Floppy Disk Controller
4. System I/O connectors mounted on the rear panel
5. 96 key keyboard
6. 3" X 5" speaker
7. Fan for maximum system reliability

COMPUTER MODULE POWER SUPPLY

The power supply was custom designed by MTU for the MTU-130. The unit has sufficient capacity to power the MONOMEG Processor, the Floppy Disk Controller, an additional 128K bytes of RAM, and two I/O expansion boards. The unit supplies +8 and +16 volts unregulated and +5, +12 and -12 volts regulated. It is designed to run on 115/220 Volts AC 60/50 Hz power and has been designed to pass both USA and International safety standards. The major features are:

1. Detachable grounded line cord for portability and international use
2. Power ON/OFF switch
3. A built-in noise/spike suppression filter
4. 2 switched AC outlets for CRT and disk drive power control
5. Primary AC fuse

INTERNAL EXPANSION CARD FILE

Inside the Computer Module is a 5 position card file frame with backplane, connectors, and card guides. The system comes with the MONOMEG board mounted in slot 1 and the Floppy Disk controller in slot 2. All connectors carry identical signals. The CPU expansion bus signals and power are distributed to each board position by a double sided, glass epoxy printed circuit backplane with 5 connectors. The connectors have gold plated contacts for long term reliable operation. There are no components other than connectors on the backplane. The frame accepts 11 inch wide boards with the following maximum depths:

POSITION 1 - 10.5 inch depth (MONOMEG processor board)
POSITION 2 - 7.5 inch depth (Floppy disk controller or Rigid disk controller)
POSITION 3 - 7.5 inch depth (Advanced display, Rigid disk controller or I/O)
POSITION 4 - 5 inch depth (128K RAM or I/O expansion)
POSITION 5 - 5 inch depth (I/O expansion)

This provides 3 spare positions (numbers 3-5) for optional MTU expansion boards or custom user designed boards. The end of this document contains descriptions of present and planned future expansion boards.

The CPU expansion bus is provided on a 44 pin (dual 22 pin) edge connector. The bus and board dimensions are basically compatible with the KIM/SYM/AIM bus which MTU has been supporting. One exception is that the MONOMEG Processor board does have bus buffers on all signal lines.

SYSTEM ELECTRONICS

The MTU-130 system is designed around the MONOMEG, a high performance 6502 microprocessor board running at 1 MegaHertz. In addition, a high performance Floppy Disk Controller board is included for control of up to 4 single or double sided, double density 8 inch Floppy drives.

The MTU-130 system has 80K of RAM for the user and the system. 64K of RAM is addressed in Bank 0 from 00000 to 0FEFF and 16K of RAM for the display is addressed in Bank 1 from 1C000 to 1FFFF. A software controlled write protect feature that affects addresses 08000-0BFFF is provided. All RAM memory is dynamic, but dedicated hardware refreshes the memory, giving transparent, interleaved memory operation.

In order to allow expansion, provision is made for addressing up to 256K bytes of memory through use of an 18 bit address bus and specialized hardware logic. Address space is divided into four 64K byte banks. Since display memory is in bank 1, all applications using the MONOMEG display make use of the bank switching feature.

In other systems, address space expansion schemes use bank selection performed by an I/O port or a complex segmentation scheme using block maps. The first method does not increase the amount of memory conveniently addressed by a single program; instead it is most suitable for time sharing systems where a number of relatively small, independent programs must co-exist in memory simultaneously. The block map scheme generally requires a microprocessor whose instruction set includes a segment field in all memory reference instructions and thus is not directly applicable to the 6502 or any other standard 8 bit microprocessor.

The MTU-130 system bank switching is based on the straightforward concept of separation of program and data memory. With such separation, both the program and the data can be up to 64K bytes each with little if any loss in programming ease. This is particularly applicable to interpretive languages (BASIC, PASCAL, FORTH) where the "user program" is actually data for the language interpreter. However it is also applicable to assembly language and compiler language programs which manipulate large data arrays.

The 6502 microprocessor does not provide signals to distinguish between data references and program references, but a study of its instruction set reveals that only the indirect addressing modes ((indirect,X) and (indirect),Y) are usable to access large blocks of data. All other addressing modes are therefore assumed to refer to program memory. Thus the bank switching logic keys on the addressing mode being used and steers indirect references to the current DATA BANK and all other references to the current PROGRAM BANK. Interrupt handling problems have been solved by temporarily diverting direct accesses to Bank 0 during the interrupt service and allowing the bank selection control registers to be read back. Both program and data banks can be selected under software control with automatic defaults to bank 0.

For special user needs, the system does have provisions for 16K bytes of either 2732 EPROMs or 2333 masked ROMs to make it usable in a standalone configuration without external storage devices. The sockets have a fixed address and are switched on or off as a group under software control. When the ROM sockets are switched off, RAM on the Disk Controller is switched on. Jumpers control the power-up or RESET enable of the Disk Controller RAM. The MTU-130 does not utilize these sockets, but does have a 256 byte "bootstrap loader" ROM on the Floppy Disk Controller to automatically load the operating software and selected application programs at power-up or cold reset.

CRT DISPLAY VIDEO OPERATION

The computer is designed with a flexible video CRT display system. It is capable of both high resolution black and white graphics and reduced resolution 4-level gray scale. This form of display operation allows any type of "data" display to be presented. The obvious use is for graphics and charts, but with the standard text display driver software it also functions as a text display of 25 lines of 80 characters per line. With optional software drivers, larger characters, other fonts, other languages, multiple viewport "windows", etc. can be displayed and manipulated.

The high resolution mode produces a dot matrix 480 dots wide and 256 dots high. The first byte of the display memory appears at the upper left CRT corner as a row of 8 dots, each one corresponding to a bit in the byte. 60 bytes (60 x 8 = 480) make a single horizontal "dot" line. The 61st memory byte is the leftmost eight dots of the second dot line, and so on.

The gray scale mode is similar to the black and white mode but generates a dot matrix of 240 dots wide and 256 dots high with 4 levels of brightness (black, dim, medium, bright). When in the gray scale mode, adjacent pairs of corresponding high resolution dots are combined into a single wider dot (not a 4 cell square). The gray levels are such that medium is the same video intensity as white when in the high resolution mode, giving one level dimmer and one level brighter.

The display memory is addressable for read and write operations through the system 18 bit addressing rules. Accesses made to the display through the system display software automatically control the bank switching for the user/programmer.

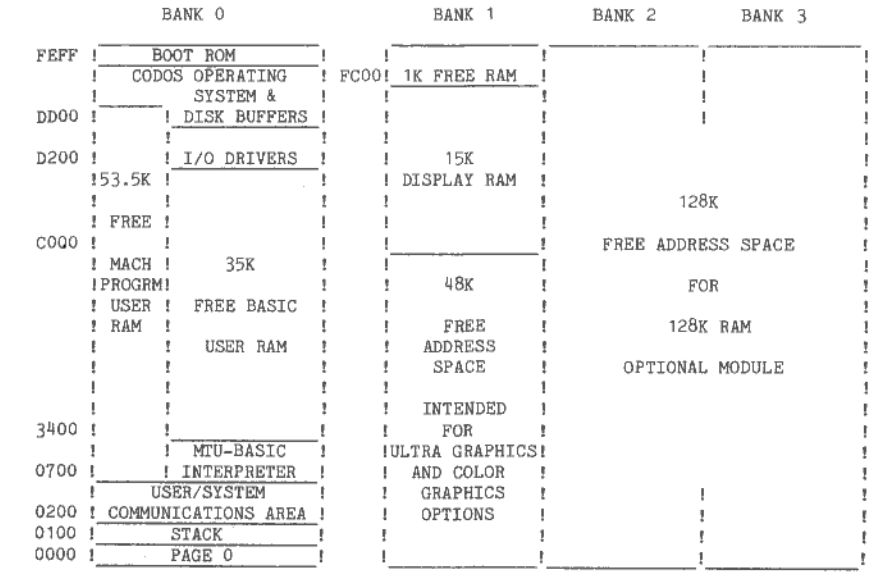
A vertical retrace signal is provided. This signal can be programmed to generate a system interrupt when the display video signal is BLANKED for vertical retrace. Thus an application program utilizing this signal can modify the CRT image memory while the display is blanked. This permits smooth graphic animation. Additionally, the signal can be used to trigger automatic keyboard scanning, elapsed time measurement or other uses.

HARDWARE SERIAL NUMBER

Each computer shipped from the MTU factory contains a unique serial number programmed into the hardware. This number can be read by any program. In addition, MTU has the ability to program custom USER/VENDOR numbers into systems. In this case, the program should check the USER NUMBER instead of the serial number. This permits protection of the software and allows the user to run the same "protected" program on all of his systems. A unique VENDOR NUMBER assigned by MTU and programmed into the hardware allows a vendor's software to only run on the systems he sells. The actual checking method used by the vendor does not need to be disclosed to MTU.

MEMORY MAPS

Shown below is the system memory map. Note that with the BASIC INTERPRETER loaded, the user has 35K free RAM. When using machine language programs, up to 53.5K of RAM is available if the standard I/O drivers and some CODOS expansion buffers are not used.



NOTE: I/O addresses occupy 0BE00-0BFFF when enabled under software control.

The following table shows the page 0 through page 6 addresses used.

ADDRESS	USAGE
06E0-06FF	TAB STOP TABLE
0600-06DF	OUTPUT LINE BUFFER
05C0-05FF	FUNCTION KEY LEGENDS
0500-05BF	INPUT LINE BUFFER
0400-04FF	FUNCTION KEY STRINGS
0300-03FF	JUMP TABLES AND SYSTEM SCRATCH AREA
0200-02FF	GLOBAL MODIFIABLE SYSTEM PARAMETERS
0110-01FF	STACK
0100-010F	PROGRAM FOR SWITCHING PROGRAM BANK AND JUMPING TO IT
00F0-00FF	I/O ROUTINES SCRATCH AREA OR FREE
00C1-00EF	CODOS USAGE OR FREE
00B0-00C0	SVC REGISTERS OR FREE
0000-00AF	----FREE

FLOPPY DISK CONTROLLER

For maximum ease of use and system speed, the MTU-130 computer was designed for Floppy Disk operation. The Floppy Disk Controller board is contained in the Computer Module along with the MONOMEG Processor. It was designed from the ground up for speed and reliability in data transfer operations. It uses an MTU designed analog phase locked loop data separator using double density operation while sustaining maximum accuracy for data storage and recall. The system design is based around the most advanced controller chip available, the NEC uPD765. This chip uses sector formating of the diskettes which allowed MTU to design a format which is maximized for retrieval speed and reliability.

The NEC chip has the ability to perform Direct Memory Access (DMA) control. This fact and the location of a block of 16K bytes of memory on-board with a disk address/data port separate from the main computer address/data port, allows disk transfers to be done at very high speeds without processor involvement. Additionally, the processor is freed up to perform interrupt servicing operations while a disk operation is being performed. This further improves system speed and flexibility for the user. The on-board 256 byte "bootstrap" ROM contains software which automatically loads the system software when the power is turned on (or cold reset performed). This places the system into full operation automatically.

Using the MTU Channel Oriented Disk Operating System (CODOS), a sustained data transfer rate is achieved of 19.6K bytes per second from disk to any memory in the system. Thus, a 32K byte disk file will be loaded into memory in less than 3 seconds. Additionally, any single byte in a disk file can be retrieved in an average of 283 milliseconds and the following 19.6K bytes in the next 1 second. At power-up or cold reset the computer will automatically load and run its system software and any programs which have been specified to execute automatically (filename "STARTUP.J").

Under specialized disk format control, a sustained disk data transfer rate of 36K bytes per second can be held for an indefinite time period by swapping diskettes in the standard dual disk drives. This gives the system an incredible ability to perform heretofore experimental operations in fields such as delayed playback music or sound output, data acquisition, speech processing, etc. Using this same capability of specialized formats allows the optional DISKEX exchange utility program to read and write other diskette formats including IBM, CP/M, and FLEX. This permits transfer of text and/or data to other systems not running under CODOS.

The disk controller signals are cabled to the Computer Module rear panel where they are available at a connector. This connector mates with the standard 50 wire disk ribbon cable connectors used on the MTU disk drive packages.

All I/O and disk connnections to the Computer Module are made through connectors mounted on its rear vertical panel. These connectors are cabled internally to the appropriate electronic boards. Additionally, there are cutouts provided to mount 2 expansion connector plates. The following paragraphs describe the standard system I/O functions.

I/O SWITCHING AND INTERFACES

The computer has a standard set of Input/Output "ports" which can be used to interface with outside world devices such as printers, modems, lab equipment, etc. These ports are controlled through registers addressed at OBEXX and OBFXX. In order to provide for the maximum system flexibility, provision is made to software switch these I/O pages out and instead let the on-board RAM that it normally covers show through. This enables 63.75K of pure contiguous RAM in Bank 0. A software settable control bit enables I/O at power-up or cold reset. For maximum benefit, expansion I/O which is off the MONOMEG board is controlled by an additional signal present on the CPU expansion bus. When this signal is used in conjunction with the lower 9 address bits for address recognition and decoding, full utilization of this switching system can occur.

VIDEO OUTPUT SIGNAL - The video display signal is output through a Phono plug mounted on the rear panel. This signal has the following characteristics: Video dot frequency - 10MHz; requires a monitor bandwidth of 5MHz minimum (standard CRT monitors). Horizontal frequency - 16,129Hz. Vertical frequency - 59.959Hz. Video amplitude - 2.5 volts peak-to-peak, DC coupled with sync at +0.8 volts and maximum white at +3.3 volts. Source Impedance - 75 ohms resistive. Separate sync - TTL level video (?), horizontal sync, and vertical sync are available. The standard MTU CRT Module and most other standard CRT monitors accept this video signal.

FIBER OPTIC LIGHT PEN - The standard light pen wand and fiber optic cable mates with a panel mount connector on the case rear vertical panel. The unit can be easily disconnected if needed. Inside the case is the Light Pen Amplifier board with photodiode, which provides a clean "hit" signal latching both X and Y screen coordinates. In order to make it easier to determine when the light pen register contents are valid, a "hit" flag is provided in the I/O address section which can be read. The digitizing response speed of the pen is 60 points per second, with a resolution to a single pixel and ± 2 pixels uncertainty. MTU-BASIC IGL and VGL Enhancement Libraries add commands to BASIC to obtain the X and Y coordinates from the Light Pen.

TWO PARALLEL PORTS - Two 8 bit bidirectional ports with 4 handshake lines (full 6522 chip) are available at a 36 pin connector mounted to the rear vertical panel. Additionally, +5V, +12V, and -12V regulated power and an active low RESET signal are provided for use by peripherals tied to this connector. Both 6522 timers are free for use through software access. A third I/O port is available inside the system if other expansion options are not in use.

ONE RS-232 PORT - A single serial port conforming to a subset of the RS-232 standard is provided, which is capable of simultaneous transmit and receive (full duplex). The baud rate is software selectable for: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200. The following RS-232C signals are provided: Transmitted data, Received data, Request to send, Clear to send, Data terminal ready, Data set ready, Carrier detect. The port utilizes a 6551 chip with its IRQ output connected to the IRQ bus line allowing interrupt driven operation. A control bit in the 6551 determines whether interrupts will be generated. All signals are available at a 25 pin D-type connector dedicated to the serial port. Several system application Utility programs are provided which utilize this port, including a "dumb terminal" Utility.

8 BIT AUDIO DAC SOUND PORT - This circuit consists of an 8 bit Digital to Analog Converter with a sharp cutoff lowpass filter and 1 watt audio power amplifier on the output. This sound port is connected to a built-in 3"X 5" speaker and to an external phone jack which disconnects the internal speaker when used. A volume control is also provided on the rear panel. The system uses this port to signal the user audibly when necessary. Additional application software allows using this port for MTU digital synthesized music creation, and playback of predigitized voice or other sounds stored as "sound" files on disk.

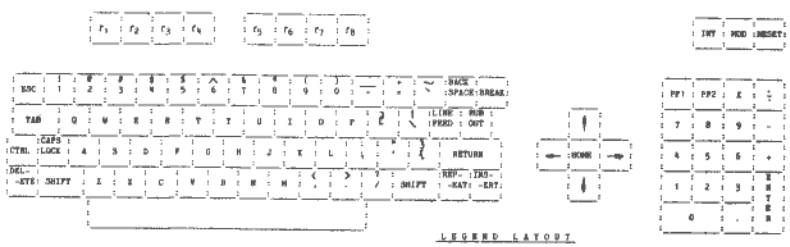
CASSETTE HARDWARE INTERFACE - This interface is provided to allow users to read tapes produced by other systems and to write tapes readable by other systems. Both cassette input and output plugs are provided on the rear panel. The cassette circuitry is capable of reading or writing most audio cassette formats known by MTU to be in general use with the use of appropriate software drivers.

NETWORK PORT - Circuitry is provided to interface to a medium speed (50K baud) interrupt driven, inter-processor local communications network option. The actual network transmit/receive circuitry, connectors and software are optional.

SYSTEM KEYBOARD

The system keyboard is designed for maximum flexibility and low cost. The keyswitch encoding is done completely in software, giving a programmer the option to reassign key functions or perform different encoding if desired. The particular keyswitch used contains 4 mechanical contacts, giving excellent reliability and extended lifetime. The following key arrays and functions are provided and can be seen in the picture below:

1. 62 key main alpha/numeric array (including a BREAK key)
2. 5 cursor keys arranged as a cross
3. 18 key function pad with numeric/calculator legends + 2 function keys
4. 8 (two 1x4) soft legend programmable function keys
5. 3 control keys - INTERRUPT, MODIFY, RESET (warm/cold reset & NMI)
6. POWER ON indicator



LEGEND LAYOUT

There are 10 user programmable function keys provided. Eight of these are fully supported by high level languages and CODOS. Their "TITLES" are displayed on the 25th line on the CRT (bottom line) enclosed in 8 rectangular boxes. Their functions can change dynamically under software control, reducing the need to "remember" what functions can be done in which "modes".

The keyboard interfaces directly to the MONOMEG through a connecting cable and is operated in a polled mode (interrupt is possible with a custom user routine and a system timer). Each key is encoded into ASCII codes for the standard keys and additional codes for special MTU keys. All 128 ASCII codes can be generated. Full keyswitch debouncing is provided. 2-key rollover encoding is used which handles multiple key depressions in a predictable manner. Using the REPEAT key, all keys may be made to repeat. In addition, the 4 cursor direction keys, plus SPACE, BACKSPACE, RUBOUT and DELETE character have automatic "timed out" repeat if held down for a minimum time period. The repeat rate is resettable by the user. Acoustic feedback is provided for each keystroke, which can be turned off by the user.

Several levels of interaction with the system hardware and software are provided for easy interrupting and resetting from the Keyboard. The RESET key initiates a warm reset (does not clear memory or re-boot the operating system). Depression of MODIFY and RESET together initiates a cold reset (same as power-on). Depression of INTERUPT halts the program which is running and returns control to the system program. The BREAK key is normally inactive but can be activated by an application program to interrupt the program when depressed. We feel that all users will come to appreciate these functions.

CRT MODULE

In the MTU-130-XX packages, the CRT Module is supplied when purchased from the factory. If purchased from a dealer, the supplied module may be slightly different. The unit contains a 12" diagonal, P-31 short persistence green phosphor screen enclosed in a plastic TV style housing. The video amplifier has a fast risetime creating a sharp, well focused image on the screen. The unit has a very slight fixed upward angle on the screen. Power required is 110 VAC, 60 Hz and can plug into one of the switched AC outlets on the MTU-130 case back. Three front panel controls are available: Horizontal Hold, Contrast, and Brightness. Additionally, this Module includes a video cable to plug into the MTU-130 video output connector.

DISK MODULE

All MTU-130-XX systems come supplied with a Disk Module containing either one or two 8" floppy disk drives. The following versions are available:

SYSTEM NO.	DRIVE TYPE	DRIVES	STORAGE CAPACITY
MTU-130-1S	single sided	1	500,000 bytes
MTU-130-1D	double sided	1	1,000,000 bytes
MTU-130-2S	single sided	2	1,000,000 bytes
MTU-130-2D	double sided	2	2,000,000 bytes

The Module mounts the drives with the diskette insertion slot vertical. The package contains room for 2 drives in all Module versions. The drives are separated by a storage area for diskettes. The unit contains power supply, fan and internal cables for 2 drives.

The standard drives supplied are from Shugart. Necessary jumper settings and testing have been done to ensure proper operation with the MTU-130 system. MTU reserves the right to substitute drives from other manufacturers. For special applications, MTU can assemble 2 DISK MODULES with a single cable for up to 4 Megabytes of on-line storage. Users who supply their own disk drive subsystem will have to jumper their drives for proper operation.

3.0 MTU-130 SOFTWARE SPECIFICATIONS

This section describes the standard software supplied with the MTU-130 and additionally available optional software. Reference to any expected future software is not to be interpreted as a firm company commitment. The evolution of MTU-130 software will continue at MTU, at other software design houses, and through customer contributions. MTU is forming an MTU USER GROUP (MUG) to enhance the exchange of user generated software and MTU "factory" information. Anyone interested in being involved in this group, please write MTU and let us know. The list below gives the standard and optional software available for the MTU-130 system as well as future expected software.

STANDARD - MTU-130 standard supplied at no additional cost

CODOS Version 2.0 with integrated I/O drivers and Utilities
MTU-EDITOR - A very powerful full screen infinite scrolling editor
Realtime digital synthesized music (MTU simplified version)
Demonstration diskette

OPTIONAL - MTU-130 optional software available at extra cost

MTU-BASIC VERSION 1.0 - MICROSOFT/MTU interpretive BASIC including:
- Integer Graphic Library
- Virtual Graphic Library
- CODOS Interface Library

Keyword Graphic Library
MTU-ASSEMBLER - MTU Disk based ASSEMBLER/DISASSEMBLER
DISKEX - Disk Exchange Utility Version 1.0
MTU-MUSIC - real time INSTRUMENT synthesis music software

FUTURE - Expected future software available in first half of 1982

FORTH language
PASCAL language with graphic extensions
Cassette tape read/write UTILITY
PET and APPLE BASIC to MTU-BASIC translator
Text/Graphic Processor based on MTU-EDITOR
Music editor with score entry and editing
Non-realtime computed, delayed playback polyphonic music system

CODOS 2.0 OPERATING SYSTEM

CODOS (Channel-Oriented Disk Operating System) is the standard operating system for the MTU-130 system. It is automatically loaded at power-up or RESET by the PROM bootstrap loader. CODOS is designed to be used by both neophyte and sophisticated users. It performs all necessary housekeeping functions for disk files while automatically maintaining an up-to-date, redundantly recorded directory. All operations under CODOS are performed via "I/O channels". These channels may be assigned to disk files, console or other input or output devices and operations may thus take place directly between the assigned channels. CODOS supports from one to four disk drives with any combination of single or double sided drives.

A monitor program is included in the CODOS system with 34 built-in commands. In order to conserve memory space while providing the functionality of a large system, some of these commands execute as overlays, loaded automatically by the system when needed. All error messages are in English, such as "SPECIFIED FILE WAS NOT FOUND", and are read from disk when needed. User-defined commands may be easily added which are executed by typing the associated file name as a command.

All types of commands can be read from disk files or other devices as if they had been typed on the keyboard. This allows a "batch" job to be performed which might consist of several programs executed in sequence. A special "STARTUP" file is automatically read and executed by the Monitor when CODOS is "booted" up. This allows automatic loading of customized device drivers, turnkey application programs, etc.

CODOS includes support for bank switching and support for MTU-BASIC language. The bank switching enhancements allow reading or writing into any memory bank from the disk. Support for BASIC includes cold & warm entry, keyboard and display handling, and user-defined commands written in BASIC. If a user defined command is a machine language program it loads as in the present system. If it is a BASIC file, BASIC autoloads (if not presently loaded), then the program is loaded and is configured with any Library packages needed to run that program (called by the LIB command).

Machine language program interfacing to CODOS is simplified through the use of address independent SuperVisor Call (SVC) instructions. The SVC consists of a 6502 BRK instruction followed by a one byte code indicating which system function is to be performed. The SVC processor does not interfere with interrupts. Because CODOS does not use any interrupts or critically-timed loops, application programs are free to fully utilize interrupts, even during disk accesses.

SVC#	FUNCTION	SVC#	FUNCTION
0	Exit to CODOS Monitor	15	Read record from channel
1	Exit to monitor with message	16	Write record to channel
2	Output inline text message to chan'l	17	Change file pos. to Begin-of-Data
3	Input byte from channel	18	Change file pos. to End-of-File
4	Output byte to channel	19	Specify file position
5	Input line from channel	20	Querry file position
6	Output line to channel	21	Assign channel to file or device
7	Output string to channel	22	Free channel
8	Convert ASCII hex to value	23	Truncate file at present position
9	Convert ASCII decimal to value	24	Define interrupt vector
10	Convert value to ASCII hex	25	Define error-recovery vector
11	Convert value to ASCII decimal	26	Restore default error recovery
12	Querry default I-O line buffers loc.	27	Enter 16-bit Pseudo-processor
13	Execute CODOS monitor command	28	Querry CODOS Version
14	Querry channel assignment	29	Querry file status

BUILT-IN MONITOR COMMANDS

ASSIGN	Display or alter an I-O channel assignment
BEGINOF	position the file pointer to the Beginning-of-Data
BP	Breakpoint command to simplify machine language debugging
BOOT	Reload the CODOS operating system software
CLOSE	Terminate operations on a specified disk
COMPARE	Compare two blocks of memory
COPY	Copy memory block
DATE	Set the date
DELETE	Delete a disk file
DISK	Display the # of files and amount of unused space remaining on each drive
DO	Execute a list of Monitor or user commands from a "batch" job file
DRIVE	Specify the default drive number
DUMP	Display contents of memory in hex & ASCII
ENDOF	Position file pointer to the End-of-data
FILES	Display names of files on disk
FILL	Fill memory block with a constant
FREE	Release specified channel if assigned
GET	Load program or memory image(s) from file
GETLOC	Display requested file load-addresses and entry point
GO	Begin execution of program
HUNT	Search for string of bytes in memory
LOCK	Write-protect the specified file
MSG	Send a message to a designated channel
NEXT	Resume execution of suspended program
ONKEY	define function key legend and substitution string
OPEN	Open the specified disk for use
PROTECT	Enable memory write-protect
REG	Display or change CPU registers
RENAME	Change the name of a file
SAVE	Save program or memory image (s) as a disk file
SET	Set memory location(s) to specified value(s)
SVC	Enable/disable Supervisor Calls
TYPE	Display, print, copy, or append text file
UNLOCK	Disable write-protect on the specified file
UNPROTECT	Disable memory write-protect

CODOS I/O DRIVERS

These programs are embedded in CODOS, providing control of the hardware devices which the user needs or is most likely to need when using the MTU-130. The Graphic, Keyboard and Text drivers are assumed to be in operation by CODOS and all MTU system programs. Where applicable, the drivers are table driven, allowing "jumps" out to provide alternate device drivers for specialized applications.

KEYBOARD DRIVER - The console keyboard is operated in a polled mode. The encoding of each key is into ASCII codes for the standard keys and additional codes for special MTU keys. All 128 ASCII codes can be generated by the keyboard. Full key-switch debouncing is provided. Operator keyboard entry errors are reduced by 2-key rollover encoding which handles multiple key depressions in a predictable manner. All keys may be made to repeat by simultaneous depression of the desired key and the REPEAT key. The 4 cursor direction keys plus SPACE, BACKSPACE, RUBOUT, and DELETE character have automatic "timed out" repeat if held down for a minimum time period, with the repeat rate resettable by the user. Acoustic feedback is provided for each keystroke in the form of a "click" which can be turned off if desired.

TEXT DRIVER - This driver creates a standard 96 character text display as 24 text lines by 80 characters per line. The 25th line shows 8 rectangles which receive the legends for the programmable function keys. The 96 character font is created as 5 X 7 dot matrix characters in a 6 X 10 cell. Both upper and lower case characters (with decenders and a dot over the j) are supported for maximum legibility. Text can be freely interspersed with graphics. If the user desires, a custom font can be linked instead of the standard font. The text cursor is located by line number and character number with home (upper left corner) being (1,1). When 80 characters are exceeded on a single line, the system performs automatic character wrap with the 81st character on the next line down in the column 1 position (far left side).

GRAPHIC DRIVER - This driver allows the user to create graphic images on the CRT screen. The screen is described using standard rectangular coordinates, (X,Y) with positive X and positive Y coordinates and the (0,0) origin at the lower left corner of the screen. All arithmetic is done with 16 bit unsigned integer coordinates. Allowable ranges for X are 0 to 479 and for Y are 0 to 255. Coordinates plotted outside these ranges are forced to the appropriate legal extreme. Two forms of graphic input are available. The (X,Y) coordinate can be created by light pen, or by positioning a full axis (both X and Y directions) cross-hair cursor with the intersection point being the pixel (X,Y) coordinate. Both the Virtual Graphics and Integer Graphics Enhancement Libraries utilize this driver.

PRINTER DRIVERS - The standard I/O driver package includes 3 printer drivers which allow text printing. The EPSON MX-70, Paper Tiger 440, and the ANACOM 150 are supported. The first 2 printers are interfaced through a parallel I/O port and the ANACOM is interfaced through the RS-232 serial interface with handshaking using the CTS (Clear-To-Send) modem control line. For screen dump graphic printing, see the UTILITY package VMDUMP routine. Drivers for other printers can be easily adapted from these driver programs.

CODOS UTILITY PROGRAMS

The following Utility programs are supplied with CODOS at no additional cost. Utility programs differ very little from CODOS built-in commands from the user's viewpoint. Utilities are invoked from the Monitor by merely typing the name of the desired Utility followed by any required or optional arguments, just as is the case for the built-in commands. Additionally they can be called from a "JOB" file in a batch operation. However, Utilities do have the following distinctions: (1) all Utility names appear in the disk directory just like any user command, (2) the Utility programs do not execute in System RAM, but elsewhere in memory.

COPYF - Copies whole files in a single or multi-drive system. Designation of files to be copied is by filename with the use of wildcard(s). If an attempt is made to copy a specific file and that file already exists on the destination disk, an error message will be given. If one or more wildcards are used for matching, then files which match but already exist on the destination disk are simply ignored and the Utility execution continues. This allows users to easily copy all files which do not already exist on the destination disk by simply typing COPYF without arguments. An "index" of what is copied and what files already existed is shown. This command also allows duplicating a file on the same disk with a different name assigned to the new file.

FORMAT - Erases and re-formats a disk for CODOS use. The user can elect to copy the operating system and to have the utility test and bypass defective disk sectors. This permits a damaged disk to be placed back into use after erasing it clean. The disk is truly unusable if an error exists in the directory or system overlay area of the disk. If this damage is found, the Utility aborts with the message "DISK UNUSABLE". Additionally, a Volume Serial Number may be assigned to uniquely identify each diskette.

KILL - Deletes files matching a given name using "wildcard" character matching. This allows rapid deletion of files which are no longer needed on the designated diskette. A display prompt is given for operator approval before each matched file is deleted. If a file is LOCKed, it cannot be KILLed.

SYSGENDEVICE - Changes the default Input-Output device drivers in the system. The Utility allows changing the names, characteristics and memory addresses for all I/O drivers. Further, it allows addition of up to 6 custom devices besides the null device and the Keyboard and Display, such as printers, tape readers, modems, etc.

SYSGENDISK - Permits modification of alterable disk drive attributes which are: (1) The number of disks in the system, (2) The number of disk buffers in the system, (3) The disk drive track-to-track step time, (4) The disk drive head load time. Any copies of the system disk made using FORMAT after running SYSGENDISK will also have the new attributes.

BROWSE UTILITY - Permits fast and efficient random scanning of large CODOS text files without using the MTU-EDITOR. An in-memory index to each line of the file is made for instant random access. The scan rate is 10,000 bytes/second to create the index with the index capacity being 10,000 text lines. Once the index is built, the user can display 20 lines starting at any line in the file by typing the desired line number. If a signed (+ or -) number is entered, BROWSE will skip forward or backward through the file using the specified displacement. If a character string in quotes is entered, BROWSE will search for that string starting at the current line number.

DIR - Displays the file INDEX attributes of selected files. The files to be seen are selected by filename or optionally using "wildcard" characters. The information displayed is: filename creation date, size, and protection status.

UPLOAD - Transmits a text file through the serial port to an external device. The baud rate is 4800 baud (alterable by patching the program). An exact image of the file specified as the argument is transmitted. XON-XOFF protocol for handshaking is implemented with an ASCII ETX ($03) sent when the end-of-file is reached.

DOWNLOAD - Inputs a text file through the serial port from an external device to a specified disk file. The baud rate is 4800 baud with XON-XOFF protocol handshaking implemented. Receipt of an ASCII ETX ($03) or a data flow interruption of more than 3 seconds when the line is in an XON state terminates the file.

SERLDR - Loads MOS Technology standard ASCII HEX object code into memory from the serial port. The transmission baud rate is 4800 baud. An optional load address can be specified to override the load address specified on the object code module. Loading can be cancelled at any time with CNTL/C.

SERDMP - Dumps memory in MOS Technology standard ASCII HEX object code format through the serial port. The transmission baud rate is 4800 baud. Dump can be cancelled at any time with CNTL/C.

SEND - Transmits a specified ASCII message over the serial port. The transmission baud rate is 4800 baud (alterable by patching the program). The message to be sent must be enclosed in quotes following the SEND command.

RECEIVE - Waits until a specified character is received from the serial port. The receiving baud rate is 4800 baud (alterable by patching the program). The wait may be terminated at any time with a CNTL/C.

WAIT - Waits for a specified number of seconds before returning control to CODOS. The wait may be terminated at any time with a CNTL/C.

STRIP - Strips linefeeds and nulls from a text file to make it compatible with the MTU-EDITOR and MTU-ASSEMBLER. Source and destination files may be specified by arguments, otherwise channel 5 is the source and channel 6 is the destination.

BACKUP - Permits fast and easy backup of CODOS diskettes. It requires a two drive system and it can backup both single and double sided diskettes.

DISKETTE - Permits direct reading and writing of physical disk sectors for debugging. Working with single or double-sided diskettes, it reads or writes specified track and sector(s) to or from a specified area in memory. A detailed error analysis is printed when a sector is unreadable.

VMDUMP - Prints a dot-for-dot image of the MTU-130 display on the Epson MX-70 printer or the Paper Tiger 440 printer.

TERM - Permits the MTU-130 to emulate a "dumb" terminal. It uses the serial port in a half-duplex with echo or a full duplex mode. The port is run from interrupts to ease timing constraints. A memory buffer is utilized to accumulate characters received during time consuming screen operations such as scroll. The maximum baud rate is 19.2K baud for bursts of 32K bytes or less. Longer sustained rates are dependent on the text, but 2400 baud is possible normally, and 300 baud worst case.

MTU-EDITOR - A FULL SCREEN SCROLLING EDITOR

MTU-EDITOR is a program which permits the generation and modification of text files. These files may be programs, text, data, or other information stored as ASCII characters. The basic editing philosophy of MTU-EDITOR is "What you see on the display is what is in the file". That is, the edited material displayed on the screen is an exact image of what is in the disk file. The currently displayed portion of the file is the "window" into the file. This window can be "scrolled" forward or backward through the file or randomly positioned to any specifed line. Modifications to the file can be made by typing directly onto the window display. As characters or lines are inserted in the window, they are simultaneously inserted into the file. During editing, no requirements are placed on the user to create disk space or perform any form of system "housekeeping". Disk space is dynamically and automatically allocated as needed. MTU-EDITOR is written entirely in machine language and uses the flexible CODOS file structure to optimize editing speed.

Files can be arbitrarily large, up to the full capacity of the disk (about one million characters for a double-sided disk). Only that portion of the file which is being edited exists in memory. When the window is repositioned, the new file portion is brought into memory automatically from the disk. No special actions are necessary to "move around" in a file larger than memory, simply request the file portion desired, and it appears in the window.

The Editor divides the console display into three areas: the Command/Status Area, a 20 line Text Window Area, and the Function Key Legend Area. The Command Area is used to display prompting messages, status, and commands typed in by the user. The Text Window lines are set off from the Command Area by a full width horizontal line indicating tab stops. The window is 20 lines of 80 characters each. MTU-EDITOR normally accepts lines of up to 80 characters in length. An alternate "landscape mode" can be selected which allows lines of up to 160 characters to be edited. Editing long lines is virtually the same as editing normal length lines. The Function Key Legend Area is at the bottom of the screen and shows eight rectangles containing the function key legends. These legends display the function name currently associated with each function key and change dynamically during the editing process to keep the user informed of the present active functions.

The Editor assigns "line numbers" to each line in the file being edited. These numbers are not shown with the actual displayed line, nor do they exist in the file on disk. They are "computed" during operation. As new lines are inserted or deleted, the line numbers are adjusted automatically. The number of the current line of interest in the Text Window is shown in the status portion of the Command Status Area.

The Editor is entered from CODOS. The system prompts the user with commands to be executed, such as whether to edit an old file in-place, create a copy of an old file and edit the copy, or create a new file. The function key line displays 8 rectangular blocks containing legends such as:

TEXT FIND MARK MOVE TABSTOP UPDATE ADDFILE CANCEL

Depression of a function key causes that associated function or "mode" to be executed immediately. Some of the command operations are described below. The description is not exhaustive as there is not sufficient space here for everything.

SCREEN EDITING - Depression of the TEXT key places the cursor in the text window and places MTU-EDITOR in text mode cancelling command mode. Movement within the text window is done with the five cursor control keys. These keys position the cursor to the desired line and column. If the Cursor Down key is depressed while on the bottom line, text in the window will "scroll" up one line showing the next line in the file. If SHIFTed Cursor-Down is depressed from any position in the window, the text window is scrolled forward by 20 lines. Similarly, Cursor-Up and SHIFTed Cursor-Up can be used to scroll backwards through the file by one line or 20 line increments.

Depression of the FIND key will position the cursor on the command line and display FIND followed by the blinking cursor. If an unsigned line number is entered followed by Carriage Return, that line number in the text file will be displayed at the top of the text window with the next 20 lines below. If the line number is signed, the Editor moves the specified number of lines in the indicated direction (+ for forward, - for back). If the entry is in quotes, that string is searched for forward from the current position of the cursor until the desired string is found. When the command has completed execution, the cursor is returned to home position in the text window.

Depressing the INSERT key places the Editor in "insert mode". Any cursor key TAB, or Carriage Return can be used to escape from insert mode. Text entered while in insert mode will be inserted at the cursor position pushing any remaining text on the line to the right as each new character is entered. When the text line is full, character insert will not allow any further insertions on that line. The BRKLINE function key can be used to break the line at the cursor position, creating a new line with the text to the right. Depressing SHIFTed INSERT will insert a new blank line with the cursor at its left end. Text lines in the window are shifted to make room for the new blank line. Characters in the text window may be changed by overstrike typing, which replaces the old characters with the new.

The DELETE key can be used to remove characters or lines. Each depression of DELETE causes the character under the cursor to be deleted and the remaining characters to the right to "close up" the gap. Holding the DELETE key down will cause it to repeat, removing large sections of a line rapidly. Depressing SHIFT and DELETE will delete the entire line the cursor is on. The remaining lines in the display window will "close up" the space where the deleted line was.

Depressing the UPDATE key causes an immediate update of the disk file copy to match the memory file copy for that portion of the file in memory. Depression of the QUIT function key will also update the file.

TAB STOP OPERATION - Tab stops can be set and cleared which the TAB key (main array) will position the cursor to. Depressing the TAB STOP function key while the cursor is on any line in text mode will set a tab stop at that cursor column position. If a stop already exists there, depression of the key will eradicate that stop. If the cursor is on the second line of a 160 character line, the stops will be set for column positions 81-160. Depression of the key on the command line followed by a Carriage Return with no numbers entered will cause the system to display TAB with the present tab setting numbers following it. Executing TAB on the command line with a string of numbers following it will set the tab stops at those numeric positions clearing any previous tab stops.

DESIGNATING TEXT - The MARK key is used to "designate" a group of lines. When the key is depressed, a secondary legend menu is displayed for the function keys which allows designating line groups and later manipulating them. To MARK text, the cursor is positioned to either the first or the last line of the desired group and the MARK key depressed. The line selected will be shown in reverse video. Repositioning the cursor and depressing MARK at the new line will designate the second boundary. That and all intervening lines are now shown in reverse video. This group of lines is now designated for further manipulation. The window can be moved between marks to designate large blocks of lines, even beyond memory size. The CANCEL key will terminate all "marking" and return the text from reverse video to normal video status.

MANIPULATING TEXT - A marked block of lines can be moved, copied, deleted or written to another file. The MOVE key moves any "marked" group of lines to the present cursor location. The lines are removed from their previous position in the file. The COPY key duplicates the "marked" group at the new cursor location canceling the marking of the original group. The DELETE key calls for verification prior to deleting the entire "marked" group. The remaining lines in the display window will "close up" the space where the deleted group was. The MTU-EDITOR automatically renumbers the text "line numbers" in all the above cases. The WRITE key followed by a file name writes the marked group in the current file to a different specified file. The command also allows "writing" to a printer or other device instead of a file.

FILE ADDING AND APPENDING - Text from other specified files may be added or inserted into the current file being edited. The file size makes no difference as long as the final combined current file fits on the disk. Depressing the INSFILE key followed by a filename will copy the entire specified file into the current file being edited. Text from the specified file is copied and inserted starting at the line the cursor is on. The ADDFILE key followed by a filename will copy the entire specified file into the file open for editing. Text from the specified file is copied at the end of the current file.

SEARCH AND REPLACE - A powerful search and replace capability is included. Depressing the CHANGE key will position the cursor on the command line and display CHANGE followed by the blinking cursor. Various parameters can be inserted following the command. This command allows global or limited search-and replace operations. It operates in a manner similar to FIND except that character strings can be replaced. The CHANGE command syntax is:

CHANGE repeat 'string 1' 'string 2' VETO 'line number'

REPEAT is optional, and gives the number of times the change is be done, defaulting to 1. Only 'string 1' is required and is the character string searched for and changed. If 'string 2' (the "replace with" string) is ommitted, then 'string 1' will simply be deleted. If the word VETO is present following the string(s) the Editor will display each string for approval before the replacement is made. The optional LINE NUMBER can be used to specify the maximum line number to be searched up to.

FOUR-VOICE MUSIC SOFTWARE

This software package allows the creation and playing of multi-part music as described in an article by Hal Chamberlin in the September 1977 issue of Byte. The musical tones produced have a limitless variety of different waveforms but the rectangular envelopes tend to give an organ-like timbre.

Using the Original Music Interpreter, each musical "event" requires as little as 1 byte of storage. Provisions for separate waveforms for each voice, variations in tempo, changes in timbre, and "musical subroutines" are available. It accepts song table code generated by the Original Notran Compiler or by hand.

The MTU-EDITOR may be used to create a songtable in ASCII character format and store it as a CODOS file. Using the Original Notran Compiler, the file can then be compiled into interpretive code for the Original Music Interpreter and output as a file to disk. The interpretive code file can then be "played" by the Original Music Interpreter.

The Original Notran Compiler accepts ASCII format songtables and compiles them into interpretive code for the Original Interpreter. Notran receives its source files from disk and saves its compiled code back to disk. A Fourier Series subroutine is built into the compiler which accepts a table of up to 16 harmonic amplitudes and phases generated by the user with MTU-EDITOR. It computes a 256 point waveform table usable with the realtime interpreter. This allows up to 16 new waveform tables (i.e. timbres) to be generated by the user. Examples of some of the encoding commands are shown below:

NVC X	DEFINE NUMBER OF VOICES TO MIX, X IS IN RANGE OF 1-4
WAV X,Y	ASSIGN WAVEFORM X TO VOICE Y. X IN RANGE OF 1-16, Y IN RANGE OF 1-4
TPO X	SET TEMPO TO X, X IN RANGE OF 1-255 QUARTER NOTE DURATION
JSR X	JUMP TO MUSICAL SUBROUTINE TO THE LINE WHOSE IDENTIFIER MATCHES X
RTS	RETURN FROM MUSICAL SUBROUTINE

SONGTABLE CODING - (OPTIONAL VOICE DIGIT) (NOTE LETTER) (OPTIONAL # OR @) (OPTIONAL OCTAVE NUMBER) (DURATION LETTER) (OPTIONAL DURATION MODIFIER)

EXAMPLE: 4C#6H.

VOICE=4TH, NOTE=C SHARP(#), 6TH OCTAVE, DOTTED HALF NOTE DURATION (H.)

SYSTEM DEMONSTRATION PROGRAMS

The following demonstration programs are supplied standard. They are written in either BASIC (B) or Assembly Language (AL) and are set up for "load and go" operation.

MUSIC - (AL) This demo plays 4 voice music through the DAC sound system.
SWIRL - (AL) This generates graphic spiral and spiderweb patterns.
LIFE - (AL) This executes the rules of the "LIFE" game on any graphic image.
CHECKER - (AL) This generates random checkerboard patterns for black/white and gray
DATAIN - (AL) Illustrates fast, interrupt driven data acquisition using CODOS.
IGLDEMO - (B) Illustrates the capabilities of the IGL BASIC Enhancement Library.
GRIN - (B) This is a simple user interactive drawing program using the GRIN cursor.
SPEECH - (AL) Illustrates that high-quality speech may be read from disk and output through the standard 8 bit DAC. Several words and phrases are stored on the disk.
SHOOT - (B) A Graphics game program which draws two cannons separated by a mountain with randomly determined dimensions and a random wind velocity and direction. Two opponents try to knock out each other's cannon.

NOTE: The following software packages are available optionally for additional cost.

MTU-BASIC INTERPRETIVE LANGUAGE

The following describes the commands that are available with the BASIC language interpreter for the MTU-130 series computers: MTU-BASIC. MTU has purchased source code rights from MICROSOFT for their 6502 implementation of the BASIC programming language, which MTU has substantially enhanced. This interpreter has the following characteristics:

1. All arithmetic is performed using floating point. Numbers are stored in binary floating point format with a 32 bit mantissa, 7 bit exponent, and sign. This gives a usable precision of 9 decimal digits, a range of $\pm 10^{**}38$.
2. Variables may consist of two characters the first of which must be alphabetic. Additional characters in the variable name are ignored.
3. Arrays may have any number of dimensions. Variable length string variables and arrays of string variables are provided. Individual strings may be up to 255 characters long.
4. Print formatting functions provided are: automatic formatting, start at given column number, skip N spaces, new line. Version 2.0 will add PRINT ... USING.

EXPANSION OF STANDARD MTU-BASIC COMMANDS - MTU-BASIC Version 1.0 is capable of linking with external Command Enhancement Libraries. Linking a Library adds an additional set of commands or functions to the already existing set. Up to 8 Libraries may be linked in simultaneously. Existing Libraries include, CODOS INTERFACE, INTEGER GRAPHICS, VIRTUAL GRAPHICS and KEYWORD GRAPHICS. Commands of one Library can supercede those of another. For example, a Library could provide a more powerful PRINT command to supercede the one in standard BASIC. During execution, BASIC verifies that the required library is actually linked and generates an appropriate error message if it is not.

This system provides a flexible way of making MTU-BASIC extensible and upward compatible. Anticipation of special commands for future applications is no longer necessary as special cases can be handled when the need arises. The method of generating linkable software will be documented for MTU-100 series users so far as it does not violate our non-disclosure agreement with Microsoft.

Additionally, MTU-BASIC Version 2 works with Bank Switching permitting use of optional memory expansion. With the use of bank switching, a BASIC program with strings and its variables or data arrays may occupy different banks available in the system. With the 128K RAM board option, this gives the BASIC programmer 64K bytes of memory for program/strings area and 64K bytes for variables/data arrays. The additional memory is usable with some minor operating restrictions. The programmer may declare the assignment of bank usage directly, or automatic assignment may be performed. A second planned enhancement for Version 2 is the addition of a PRINT...USING command for output formating.

STRING FUNCTIONS	ARITHMETIC	RELATIONAL OPERATORS	BOOLEAN OPERATORS	STRING OPERATORS	RESERVED VARIABLES
ASC					
CHR$	=	=	AND	+	ST
LEFT$	+		OR		KE(Y)
LEN	-		NOT	.	
MID$	*				
RIGHT$	/	=			
STR$	'	=			
VAL					

PROGRAM STATEMENTS	COMMANDS	I/O STATEMENTS AND FUNCTIONS	ARITHMETIC FUNCTIONS
DEF	* BYE		
DIM	CLEAR	DATA	ABS
END	CONT	GET	ATN
FOR..NEXT	* ENTER	INPUT	COS
GOSUB	* FRELIB	PRINT	EXP
GOTO	* LIB	READ	INT
IF..THEN (or GOTO)	* LEGEND	SPC	LOG
LET	LIST	TAB	RND
ON..GOTO (or GOSUB)	LOAD	POS	SGN
REM	* MCALL	* OUTCHAN	SIN
RESTORE	NEW	FRE	SQR
RETURN	POKE	PEEK	TAN
STOP	RUN		USR
WAIT	SAVE		

* MTU original commands, not in standard MICROSOFT BASIC.

IMPORTANT COMMAND DESCRIPTIONS

INPUT and GET - These commands normally receive data from the keyboard. If CODOS channel 1 has been assigned to a device other than the console keyboard (such as a job file on disk), these commands get their data from that "device". In addition, the INPUT command has been upgraded so that if a flag is set, a null response (CR only) will return a null string instead of halting the program. The default setting of this flag is for non-halting operation.

LOAD and SAVE - The LOAD command recalls the specified file from disk and loads it into the bottom of available BASIC memory. The SAVE command stores the BASIC program currently in memory to the disk under the specified file name. The lines are recalled or stored in a compact memory image format ("tokenized"), not in ASCII form.

LIST and ENTER - The LIST and ENTER commands are like the SAVE and LOAD commands except that ASCII data is written or read instead of the tokenized memory image format. The LIST command writes to the specified file on disk. If no range of lines is specified, it defaults to the whole program. If no file name is given, the listing will be displayed on the console instead of going to disk. The ENTER command reads the specified ASCII file on disk and adds the lines into the program currently in memory. This command is not found on other systems using Microsoft BASIC. Having the ENTER command allows BASIC programs to be transported in ASCII format to other systems, as opposed to the machine dependent memory image format of the SAVE and LOAD commands. Also, with LIST and ENTER you can save parts of programs and add those parts to other programs later.

LEGEND - This command allows legending of the programmable function keys from BASIC. the 25th line of the CRT. Multiple legends can be created with a single command. The first new legend starts at the specified function key number and all the remaining keys to the right are relegended or their legend cleared. Along with the KEY variable, this command allows use of the function keys under control of a BASIC program.

KEY VARIABLE - This is a new reserved variable. Its operation is ideally suited for use in ON..GOTO or ON..GOSUB statements. This makes function key driven application programs simple to implememt.

BYE - This command returns control to the CODOS operating system from BASIC.

LIB and FRELIB - The LIB (LIBrary) command is used to add Command Libraries and the FRELIB command is used to delete them from the standard MTU-BASIC command set. Filenames following LIB are assumed to be Command Libraries. LIB checks each filename in the list to see if that Library is already linked. If it is, the command will continue with the next filename in the list. If it is not, that Library will be loaded and linked to MTU-BASIC. If a filename is not a legal name (not available on the disk), then an error message is printed. If LIB is not followed by any filename parameters, the names of the currently linked Libraries will be printed out. Executing a FRELIB command will unlink all Libraries and the top of memory will be reset to the original top of memory. Both the user and BASIC programs can declare what Libraries are to be linked.

STANDARD MTU-BASIC LIBRARIES

Three Command Enhancement Libraries come standard with MTU-BASIC. The added commands may be executed from the console in immediate mode or from a running program. As with regular BASIC commands, the added commands may use expressions, variables and strings as arguments as well as constants and literals. Integer Graphics is a minimum size Enhancement Library that works in terms of integer (X,Y) screen coordinates. Virtual Graphics is a larger, more comprehensive Enhancement Library that works in terms of floating point virtual coordinates and implements display scaling and windowing functions. The CODOS Interface Library allows full utilization of all CODOS commands to be executed from BASIC. The optional Keyword Graphics Library is a larger graphics package with more emphasis on shapes and characters than graphics though it does contain its own line and dot graphic capabilities.

VIRTUAL GRAPHICS LIBRARY (VGL) - This software provides a powerful 2-dimensional graphics capability integrated into the BASIC language. The programmer uses floating point values to specify coordinates in terms of the user's units. All linework or points which fall outside of the user's presently displayed window are automatically clipped at the window boundary. This allows a program to "blow up" the detail in a small portion of a plot to the full screen size by redefining the window before plotting the data. In addition a viewport can be called which defines what physical part of the screen will be occupied by the user's window.

Another major feature of VGL is the "GRaphic INput" command. When this command is executed, a pulsating graphic crosshair cursor appears on the screen (a full-screen horizontal and vertical line). The intersection of these two lines is the current graphic cursor position, and can be "steered" to any point on the screen with the four cursor controls. Depressing any non-cursor key makes the crosshairs disappear. The BASIC program can then use the coordinates "picked" and the character key depressed. Again, the coordinates are given in terms of the user's data coordinate system. VGL also allows relative coordinates which are especially useful when an element of a picture is to be repeated several times. Additionally, VGR allows free mixing of both text and graphics anywhere on the screen. Thus charts and graphs can be titled for better understanding. A partial list of VGL command examples is shown below:

VGL COMMAND EXAMPLES

```
WINDOW -1.56,0,+1.56,100        RDRAW   11.2,9
MOVE 0,Y2                       RPEN    X4,Y5
DRAW 2*X-1, Y2                  RPOINT  -X,-Y
POINT 7.4,8-Y                   GRIN T$, XC, YC
PEN X,Y                         LABEL B$
PENMODE 0                       XAXIS YI,.1,0,10
VIEWPORT 50,XT, 400,YT          YAXIS 10,DY,20,Y9
RMOVE   0,10                    SPRINT
```

INTEGER GRAPHICS LIBRARY (IGL) - This software is similar to VGL except that it works with integer arithmetic and does not include the depth of graphic functions. IGL requires only 0.5K of RAM for operation and is a compatible subset of the VGL command set. Coordinates may be 0-479 for X and 0-255 for Y with no scaling. Some of the commands are shown below:

SCLEAR - clears the entire screen including legend area
SGRIN - allows coordinate input using the GRIN cursor.
SLPEN - allows coordinate input using the light pen.
SMOVE X,Y - moves graphics cursor to X,Y
SDRAW X,Y - draws line according to current mode from cursor to X,Y
PENMODE M - sets drawing mode (white, black, flip) according to M
TCURSOR R,C - moves the text cursor to row R column C
TWINDO T,N - sets scrolling text window top to Y=T and text lines to N

CODOS INTERFACE LIBRARY (CIL) - This enhancement package provides a convenient and flexible method for BASIC programs to read and write CODOS data files either randomly or sequentially. Using the BLKWRT and BLKRD commands, data can be written or read from a channel at CODOS speeds (read this as FAST). A flag is available to allow the INPUT command to read commas as part of a string. Additionally, using the SYSTEM command, all CODOS commands may be accessed. Some of the commands are shown below:

APPEND - Appends a saved disk program to the program in memory.
BLKRD@ - Reads data from a specified channel into specified variables or arrays using record I/O for high speed.
BLKWRT@ - Writes specified variables or arrays in memory image format (such as data arrays) to a specified channel using record I/O for high speed.
ENTER@ - Reads BASIC program statements in ASCII from specified channel.
GET@ - Reads a single byte from a specified channel.
INPUT@ - Specifies a CODOS channel to read ASCII text from.
LIST@ - Lists program to a specified channel.
ONERR@ - Designates line number/command to execute for error recovery.
POSITION@- Positions pointer for file assigned to a specified channel.
POSN( ) - Reads the position of the file assigned to a specified channel.
PRINT@ - Specifies a CODOS channel to receive ASCII text being printed.
PUT@ - Writes a single byte to a specified channel.
SYSTEM - Followed by a CODOS monitor command executes the command without returning to CODOS. Can also be executed by a running program.
TRUNC@ - Truncates the file assigned to a specified channel.

KEYWORD GRAPHICS LIBRARY (KGL) Optional - The Keyword Graphic Library is a general purpose graphics package. There are several features of KGL not available in the IGL and VGL Libraries. First of all, KGL supports the definition of 4 separate "window" areas which may be any portion of the display. Drawing and printing is restricted to the current window. Second, KGL has its own internal character drawing capability. This subroutine interprets special drawing codes that defines an object's shape. A set of such shape definitions is included with KGL that provide the complete ASCII characters. Thirdly, KGL contains a powerful Block Move capability which allows flexible repositioning of graphic areas or scrolling of text areas.

MTU-ASSEMBLER

MTU-ASSEMBLER is a disk based standard two-pass resident assembler which runs on the MTU-130. It accepts 6502 assembly language source programs as input and produces 6502 machine language programs as output along with a formatted listing. The source program to be assembled is a standard ASCII text file on disk, which is usually prepared using the MTU-EDITOR. The machine language output (object code) is written to a disk file as a standard CODOS-format program which can be loaded and executed as a standard CODOS command. The listing can be output directly to a printer or other device during assembly, or can be saved on a disk file for subsequent examination or printing.

The Assembler accepts standard MOS Technology format source code and is written entirely in machine language, optimized for speed of assembly. It is quite practical to use the MTU-130 as a development system for large assembly-language programs, since programs with thousands of lines and many hundreds of identifiers can be assembled in only a couple of minutes. Because the Assembler outputs the object code to a disk file, there are no "reserved" areas of memory where programs cannot be assembled. Any number of non-contiguous blocks of object code or data can be produced, and all will be saved on a single disk file which can be loaded and executed with a single CODOS command. A very large program typically assembles at about 2500 lines per minute with the listing suppressed and about 1500 lines per minute with the listing assigned to a file. The symbol table is large enough to handle about 1900 identifiers with cross-references. Files used may be as large as necessary, up to the full capacity of a disk.

Source program identifiers may contain up to 31 characters, all of which are "significant". The only reserved symbol is the single character "A", which denotes accumulator addressing mode. Numeric operands may be specified in hexadecimal, decimal, octal, or binary. All expressions are evaluated using triple precision arithmetic, allowing values in the range of -8,388,608 to +8,388,607 decimal. Source lines may be 0 to 80 characters long and may include tabs.

The listing shows each source statement, a line number, and the object code generated. Errors encountered during assembly are copied to the listing. Each line containing an error is flagged immediately underneath with a pointer to the offending portion of the line and a short English error message. A count of errors is displayed at the end of the assembly. The listing normally includes a Symbol Table and Cross Reference Map. The Symbol Table is sorted alphabetically. It consists of a list of all identifier names used in the program with their values and all source line numbers which refer to them. The Cross Reference Map can be suppressed if desired, in which case an abreviated Symbol Table is printed. The entire Table can also be suppressed if desired, or the entire listing can be suppressed.

An error summary is normally produced on the console screen. As an erroneous source line is encountered, it is displayed with an error message. The error summary can optionally be diverted to a printer, file, or other device, or suppressed if desired.

Special files called the "Definitions" file and the "Externals" file can be used to link programs to separately-assembled programs or data areas. A unique capability called "Direct definitions" is supported which permits temporary parameter redefinition without modifying the source code file.

A .READ pseudo can be used to read alternate source files during assembly. This feature allows standard libraries of routines to be used conveniently. .READs can be nested up to three files deep.

FOUR-VOICE INSTRUMENT SYNTHESIS MUSIC SOFTWARE

This software package allows the creation and playing of 4-part INSTRUMENT SOUND music as described in an article by Hal Chamberlin in the April 1980 issue of Byte. This package was developed with the help of Dr. Frank Covitz.

This program allows the user to specify an amplitude envelope for EACH harmonic in the tones used. Thus the overall amplitude and harmonic structure of the tone may vary during the duration of individual notes. When coded instrument specifications are based on published analyses of common musical instruments, the resulting sounds from this system closely resemble the analyzed instrument. The greatest power however comes from the ability to define original instrument sounds.

In the instrument coding, harmonic amplitude envelopes are specified as piecewise linear approximations to the desired smooth curves. Any number of line segments may be used to define the envelope of a harmonic and different harmonics may be defined by a different number of line segments. All of the routines needed to compute waveforms from instrument data are part of the music playing program and thus are always available. Another feature of the system is stereo capability which allows instruments to be assigned to either channel or changed at will (requires a second hardware 8 bit DAC connected to a parallel port).

Songs are coded into two separate lists of bytes in memory called the Sequence String and the Note String. The Sequence String contains commands to compute instrument waveforms, set the tempo, assign the voices to left or right stereo channels, reallocate waveform memory, set the number of voices between 1 and 4, set individual pitch offsets for each voice (allows on the spot transposition from one key to another), and play segments of the Note String. The Note String contains the actual note coding of musical "events". Each pitch byte has two extra bits which specify one of three possible attacks for the note or a sustain from the previous event. This allows proper articulation of complex, multi-part musical passages.

DISKEX - DISK EXCHANGE UTILITY

This utility allows files of text or data to be exchanged between CODOS and other floppy disk systems. Both read and write operations are permitted. To write a file in a supported format, the disk being written to must contain a named file large enough to accept the data or text being transferred. The systems supported in Version 1 include:

1. CP/M (8080, Z-80)
2. FLEX (6800)
3. IBM 3270

Version 2 support is expected to be extended to include:

1. UCSD (Pascal)
2. DEC RT-11 (LSI-11, PDP-11)

NOTE: This utility does not allow the user to RUN CP/M or other operating systems. It does allow the transfer of TEXT and DATA files (or any ASCII file) between these systems and CODOS.

PLANNED FUTURE SOFTWARE DEVELOPMENTS

The following gives the reader some idea of what is presently going on at MTU and at user sites in software development which should be available in early 1982.

FORTH LANGUAGE WITH CODOS AND GRAPHIC EXTENSIONS - One of our customers is taking the standard FORTH 79 language and fitting it to the MTU-130 system. One of the goals of this effort is to use CODOS instead of the standard FORTH DOS operation. This would speed up operation of FORTH programs tremendously when disk operations are needed. If this is done, a FORTH to CODOS disk exchange utility is expected to be provided also.

PASCAL LANGUAGE WITH GRAPHIC EXTENSIONS - MTU is working on a version of PASCAL for the MTU-130. We presently (August 1981) have SOFTECH (USCD) PASCAL running, but have not made a final decision to use this system versus some other PASCAL which could use CODOS and its built-in flexibility and speed. For instance, all disk file space allocation under CODOS is dynamic whereas SOFTECH PASCAL uses static allocation. CODOS makes the file as large as is needed and if it is added to, automatically makes the space larger.

CASSETTE READ/WRITE UTILITY - MTU is working on a series of cassette read/write utilities for at least PET, AIM, and KIM tape formats. This will allow users or software vendors to easily transfer their own programs over to our system via tape reading.

PET AND APPLE BASIC TO MTU-BASIC TRANSLATOR UTILITY - A present MTU user is in the process of creating a Utility which will take a PET or APPLE BASIC program in memory image and perform translation to MTU-BASIC memory image form. Not all commands can be translated, but in those conditions a note will be flagged that the user must aid in the translation.

FULL TEXT PROCESSOR PROGRAM - One of the major developments progressing is a full scale text processor program. This program continues the MTU "WHAT YOU SEE IS WHAT YOU GET" philosophy. There is very little guessing as to what the final output document will look like. Because of the flexibility of the display and disk in the MTU-130 system, features not found on standard dedicated word processors will be available. Very high emphasis is being placed on the human interface, ease of use, and transparent system operations. With the complement of features, there should be very few documents which could not be handled by this system.

SECTION 4.0 - HARDWARE/SOFTWARE EXPANSION OPTIONS

EXISTING EXPANSION PRODUCTS

PROTOTYPE BOARD - The present MTU K-1020 prototype board can be utilized in the MTU-100/130 systems for construction of custom circuits.

BANKER BOARD - The present MTU K-1032-1 and -2 BANKER memory boards can be used with the MTU-100/130 systems. Some jumpering is necessary to fully utilize the expanded I/O on the 1032-1 board with the MTU-130. These boards can be addressed for RAM Banks 0,1,2 or 3.

FUTURE EXPANSION PRODUCTS

RAM/MATH 128K RAM BOARD - This board contains 128K of RAM expansion fixed addressed at Bank 2 and 3. This board is useful with high level languages utilizing more than the standard memory provided in the basic systems. In addition, to allow high speed floating point processing and future compute power expansion, a 68000 microcomputer and interface circuitry has been designed into the board. Full interrupt and control register handshaking is provided with the 68000 accessing the on-board memory through a second memory port. It is expected that this board will allow upgrading to 256K RAM chips in the future, providing 512K bytes of RAM on-board. Control of which of the four 128K RAM blocks are coupled to the address bus (or none) is provided. Thus multiple board could be used in the system. Availability is expected in first quarter 1982.

DIGISOUND SYSTEM - This is an 8 bit Analog to Digital system with microphone, amplifier, filters and parallel I/O interface on board. This system can be used to create speech files on disk for playback under software control as speech output.

DELPLAY-12 SYSTEM - This is a high fidelity sound synthesis and analysis package containing both hardware and software. The hardware consists of a 12 bit digital to analog converter, 12 bit analog to digital converter, three sample-and-hold units, two low-pass filters, and a 1K sample FIFO buffer. A 4 bit programmable gain control extends the dynamic range to nearly 100dB. Burst 12 bit sample rates to 40KHz and continuous rates to 25KHz are possible. Expected available software will consist of direct to floppy disk sound recording and playback programs, sound analysis programs, and a powerful delayed playback music synthesis program. Analysis detail and synthesis flexibility is ample for research into the acoustics of sound and musical instruments as well.

NETWORK OPERATING SYSTEM - This hardware and software combination will allow MTU-130 units (and non-disk MTU-100 units) to communicate together over a 50K baud interrupt driven network. Five units can be connected in a continuous loop coaxial cable up to 2000 feet in length. Additional units added in reduce the loop distance by 100 feet each. Since the transmission is unidirectional, repeaters can be used to increase the distance of the loop. Various forms of software will be available from Master slave operations such as in a classroom to mailbox and full virtual communications between systems for distributed processing. Multiple stations will be able to communicate with each other or their disk files (security permitting). Availability is expected by mid 1982.

RIGID DISK CONTROLLER - MTU expects to offer a rigid disk option in 1982 for larger data base operations. It is expected that CODOS will be the operating system, making all existing files and operations compatible.

HIGH RESOLUTION GRAPHIC DISPLAY SYSTEM - MTU is working on the specifications and study of market needs for a higher resolution display system as an expansion option to the basic MTU-130 system. The present directions are for a 4K by 4K positioning accuracy display using stroke vector (not raster scan) for both character and line drawing. This type of system offers relative merits that fit well with the MTU system approach. A 15" diagonal CRT would be expected to be part of the package, allowing continued use of the standard system CRT monitor. Text capability would be offer full page operation with upwards of 4,000 characters displayed at once.

TRADEMARKS

The following trademarks have been used in this document:

CP/M is a trademark of DIgital Research
FLEX is a trademark of Technical System Consultants
KIM, PET and CBM are trademarks of Commodore Business Machines
AIM-65 is a trademark of Rockwell International
UCSD is a trademark of the regents of The University of California
RT-11 is a trademark of Digital Equipment Corp.

CODOS, DISKEX, MTU-100, MTU-130 and MTU are trademarks of
Micro Technology Unlimited a Division of Consolidated Sciences, Inc.

ORDER FORM

MICRO TECHNOLOGY UNLIMITED
2806 HILLSBOROUGH STREET
PO BOX 12106
RALEIGH, NC USA 27605
(919) 833-1458

NAME _____

ADDRESS _____

CITY _____ STATE _____ POSTAL CODE _____

COUNTRY _____ PHONE ( ) _____

QTY	MTU PART NUMBER	DESCRIPTION	PRICE EACH	TOTAL

MASTERCARD OR VISA PAYMENT FILL OUT THE FOLLOWING:

MC____ VISA ____ EXPIRES: (DATE)_____

CARD NUMBER _____

CARDHOLDER'S SIGNATURE _____

ORDER TOTAL _____

ADD SHIPPING _____

DEPOSIT _____

COD BALANCE _____

NOTE: A deposit of $500 is required with order. Your order will be confirmed in writing by MTU and you will be given a scheduled shipping date. Deposit is not refundable on cancellation unless MTU fails to ship within 30 days of the written date sent you with confirmation of your order.

SYSTEM NO.	PRICE
MTU-130	$2,640
MTU-130-1S	$3,995
MTU-130-1D	$4,195
MTU-130-2S	$4,495
MTU-130-2D	$4,995

SHIPPING

DESCRIPTION	WEIGHT	UPS ZONE 2	3	4	5	6	7	8
COMPUTER MODULE	27 POUNDS	$ 3.50	4.16	5.09	6.12	7.61	9.21	11.03
MANUAL	5 POUNDS	$ 1.56	1.68	1.86	2.05	2.33	2.62	2.96
SINGLE DISK MODULE	34 POUNDS	$ 4.11	4.94	6.13	7.41	9.28	11.30	13.59
DOUBLE DISK MODULE	45 POUNDS	$ 5.07	6.18	7.74	9.44	11.93	14.59	17.62
CRT MODULE	19 POUNDS	$ 2.80	3.26	3.92	4.64	5.69	6.81	8.10

NOTE: ALL SHIPMENTS WILL BE MADE BY UPS IN USA AND AIR FREIGHT COLLECT ELSEWHERE.

Micro Technology Unlimited
2806 Hillsborough Street
P.O. Box 12106
Raleigh, NC 27605, U.S.A.
(919) 833-1458